[試して納得！わかるディジタル信号処理　解析編]

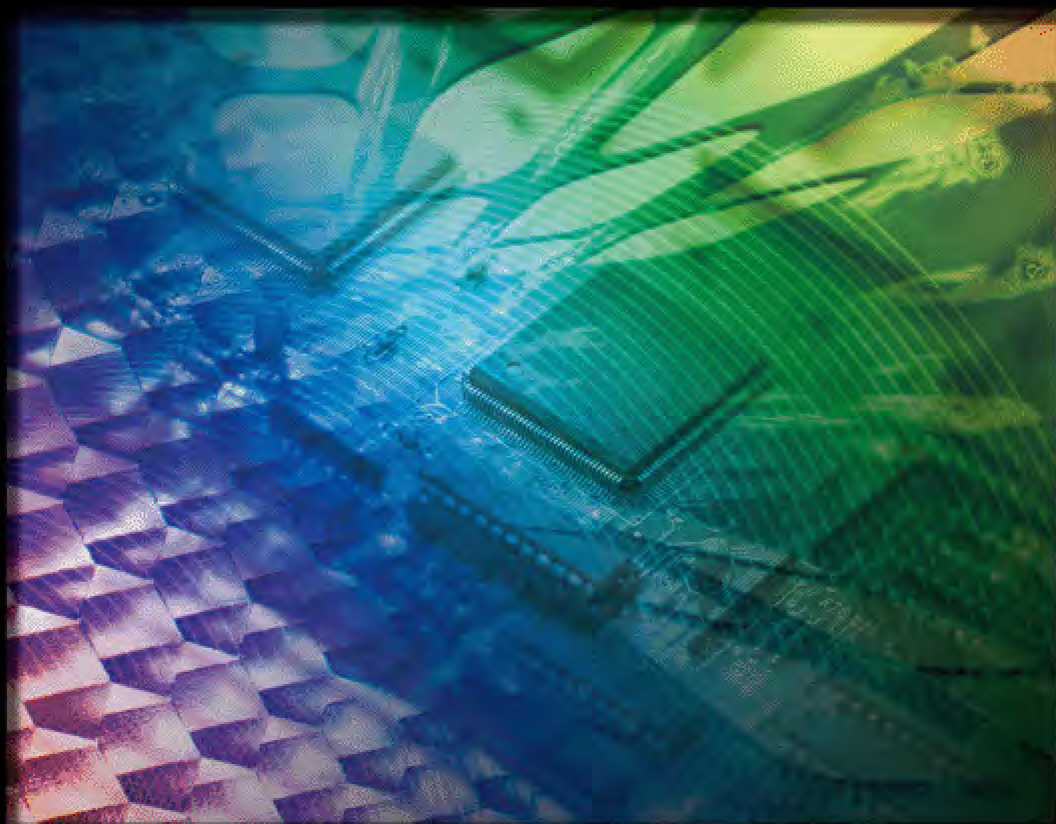
メイン　序章　第一章　第二章　第三章　第四章　付録　操作方法　実験室　各種設定の変更　本書の情報

試して納得！

IS Book
Interactive Simulation Book

わかる ディジタル信号処理 解析編

東京電機大学教授　工学博士　三谷政昭 著



最初から始める　　続きから始める

【試して納得！】シリーズ刊行にあたって

ディジタル信号処理（DSP：Digital Signal Processing）技術は，通信や音声＆画像処理，制御，計測などの多彩な分野で当たり前のように使用される要素技術なので，今こそ，この技術をしっかりと使いこなせる専門家が求められています．

最近，学生や社会人の方々から，「DSPの基本的な原理を体系的にまとめた解説書があれば助かるのになあ」という声をよく耳にします．そんな声にお応えすべく，DSPの基本を体系的にまとめ，徹底してわかりやすく，インタラクティブ（Interactive，対話的）にシミュレーション（Simulation）しながら，"楽しく読めて，目で見て，直感的にわかる"ソフトを世に出したいものだと常々考えていました．

ところで，2005年9月に発刊した電子回路向け【楽して学ぶ】シリーズは，"Interactive Simulation Book"略して『IS Book（アイエス・ブックと読む）』と称するソフトであり，世界で初めてのシミュレータ内蔵「電子教科書＆参考書」です．そこで，電子回路向けの【楽して学ぶ】シリーズのコンセプトを踏襲し，DSP解説書の実現に向けて，新たなシミュレーション・ソフトとして，【DSPシミュレータ・ソフト「商品名ＤＳＰアナライザ（DspAnalyzer）」】が開発されました．そうして，株式会社マイクロネット（浜三弘 社長）との産学連携により，DSPシミュレータを内蔵した『IS Book』として，新シリーズ【試して納得！】の発刊にこぎつけた次第です．

少々手前みそですが，DSPがどんなものなのかをすばやく知って，即座に活用するための"特効薬"となるように，「今日から使える，使いこなす，使いこなせる」ための基礎を小気味よく解説してあります．

「開けてビックリ！！　玉手箱」じゃあないけれど，本シリーズのテキスト（CD−ROM）をパソコンに突っ込んで，スイッチオン．すると，どうでしょう．仕掛け絵本のように，回路図，回路部品，工作道具，オシロスコープなどが飛び出してきます．そうして，テキストの説明を読みながら，DSPのリアルタイム・シミュレーション実験が体感できて，基礎から応用までを習得できるように，数多くの工夫が凝らしてあります．

（1）数式の使用をできるだけ避けること

数式は一つの言葉なので，物理的なイメージと結びつけることが大切です．ただやみくもに数式を暗記するだけでは，内容がさっぱりわからないというジレンマに陥ってしまいます（いわゆる，「理数離れ」症候群）．そのため，数式の表現力に頼ることをできるだけ避けて，数式を物理的な言葉で"翻訳"した表現を心がけ，みなさんの「数式に対するアレルギー」を取り去ってもらいます．そうして，直感的な理解，イメージをみなさんに植え付けます．なぜなら，物の本質の理解には順序だった（へ?）理屈も大切ですが，これ以上に重要なものは「直感的な理解，イメージ」なのですから（筆者の経験から言えることですが...）．

（2）説明の順序を理解しやすい並びにすること

みなさんの理解しやすいことを目標に，いままでのDSPの参考書にありがちな内容説明の流れにとらわれず自由な形で構成しました．

（3）チェックBoxで理解したかどうかを自己評価できること

原則として見開き 1〜2ページで説明を終える形式とし，check BOX  の質問で理解度を確認できるようになっています．

本書は，DSPが初めてという人，専門書を読んではみたが難しくてどうもとっつきにくい，わかりにくいと困っている人をとくに意識して，わかりやすく解説してあります．なお，すでに勉強したことがある人でも，副読本や復習のための参考として役立ててもらえるものと思います．

また，わかりやすく系統立てて段階的に習得できるようになっていますから，しっかりと読み進んでいってもらう過程において，短期間にディジタル信号処理の基礎から応用までの必須知識をスムーズに身につけてもらえるものと確信しています．

最後まで読破したあとには，webサイトに「応用問題」があります．どのぐらいの実力がついたのか，確認ができます．サイトへ（要登録）

終わりに，【試して納得！】シリーズを読破されたみなさんには，実践的な経験を通して，ディジタル信号処理に精通した技術者として活躍されんことを期待しつつ，筆を置くことにします．

平成21年6月　　著者しるす

目次

第一章

ディジタル信号数学の気持ちを知ろう

ディジタル信号処理（ＤＳＰ：Digital Signal Processing）それ自体の計算は加減乗除（＋，－，×，÷）の単純な四則計算のかたまり，小学生の算数の範囲なので超簡単なのだが……．とは言っても，多くのＤＳＰエンジニアは，難しい数式が多く使われていて，とっつきにくいイメージをもたれているかもしれない．おそらく「数式が外国語の文章のように見えて，日本語に翻訳することができない．だから，数式を単に丸暗記するだけに留まっており，数式のもつ物理的意味がつかめない」，という感じではないかと推測される（イラスト図1-1）．

イラスト図1-1

ディジタル信号処理(DSP)を使いこなすための基礎は信号数学にあり!!

ところで，ＤＳＰを表す数式は一つの言葉なので，物理的なイメージと結びつけることが重要である．したがって，ただ闇雲に数式を暗記するだけでは，内容がさっぱりわからないというジレンマに陥ってしまう（いわゆる，ＤＳＰアレルギー症候群）．ＤＳＰの本質を理解するには，順序だった（へ？）理屈も大切であるが，それ以上に重要なのは「直感的な理解，イメージ」と断言できる（筆者の経験から言えることですが…）．本章では，そうした状況を克服すべくディジタル信号処理のココロ（心）をつかんで，数式の翻訳のノウハウを習得してもらうために，信号数学の真髄と言える「z変換」にフォーカスして，ていねいに解説する．その際，数式の表現力に頼ることをできるだけ避けて，数式を物理的な言葉で“翻訳”した表現を心がけることにする．直感的な理解とイメージをみなさんに植え付けることによって，「ディジタル信号数学に対するアレルギー」を取り去り，ディジタル信号処理の気持ちがわかるエンジニアになってもらおうというわけだ（イラスト図1-2）．さっそく，ディジタル信号を数式で表現することから始めよう．

イラスト図1-2

一の２　ディジタル信号をデルタ数列で表現する【1/2】

まず，アナログ信号のインパルス波形 $\delta(t)$，すなわち，

$$\delta(t)=\begin{cases}1 & ;t=0\\ 0 & ;t\neq 0\end{cases}$$

を考える．ここで，$k=\cdots$，-2，-1，0，1，2，$\cdots$におけるサンプル値 $\delta_k=\delta(kT)$ は，「デルタ数列（あるいは，インパルス信号）」と呼ばれるディジタル信号系列 $\{\delta_k\}_{k=-\infty}^{k=\infty}$で，これは $k=0$ のとき 1 に等しく，$k\neq 0$ のときゼロである(図1-3)．

$$\delta_k=\begin{cases}1 & ;k=0\\ 0 & ;k\neq 0\end{cases} \tag{1-5}$$

例えば，$\{\delta_{k-5}\}_{k=-\infty}^{k=\infty}$ は$k=5$ のとき 1 に等しく，$k\neq 5$ のときゼロである．

つまり，任意の整数 m に対しては，

$$\delta_{k-m}=\begin{cases}1 & ;k=m\\ 0 & ;k\neq m\end{cases} \tag{1-6}$$

となる．

以上より，ディジタル信号$\{x[k]\}_{k=-\infty}^{k=\infty}$ をデルタ数列の重ね合わせと考えれば，

$$x[k]=\cdots+x[-2]\delta_{k+2}+x[-1]\delta_{k+1}+x[0]\delta_k+x[1]\delta_{k-1}+x[2]\delta_{k-2}+\cdots$$
$$=\sum_{n=-\infty}^{\infty}x[n]\delta_{k-n} \tag{1-7}$$

となる（図1-4，実行例1-1）．なぜなら，$\{x[n]\delta_{k-n}\}_{k=-\infty}^{k=\infty}$ は $n=k$ のとき $x[k]$ に等しく，$n\neq k$ ではゼロなのだから．つまり，式(1-7)の総和においては，ある特定の k に対する項，

$$x[k]\delta_{k-k}=x[k]\delta_0=x[k]$$

を除く残りのすべての項はゼロとなる．

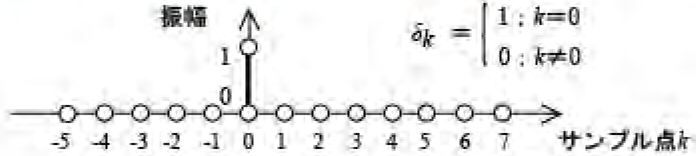


図 1-3 デルタ数列（インパルス信号）

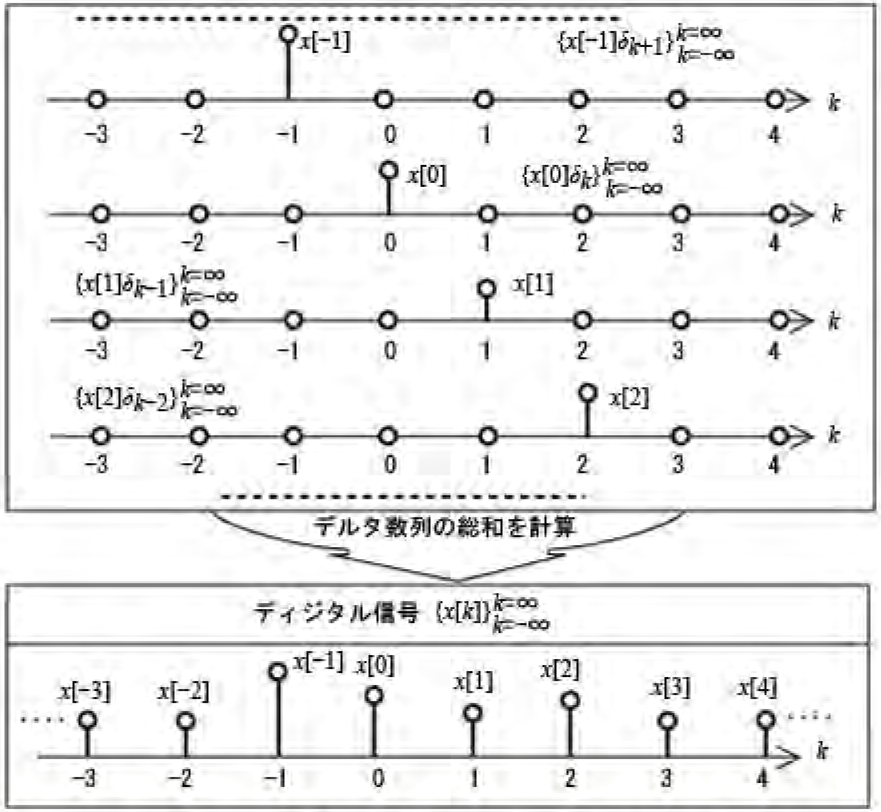


図 1-4　デルタ数列の重ね合わせとディジタル信号

一の３

z変換はディジタル信号全体を表す！【1/2】

いま，ディジタル信号のサンプル点 k（サンプリング時刻 $t=kT$）に対して，

（サンプル時刻 k）	$\cdots$,	-2,	-1,	0,	1,	2,	$\cdots$
		⇕	⇕	⇕	⇕	⇕	
（時間 t）	$\cdots$,	$-2T$,	$-T$,	0,	T,	$2T$,	$\cdots$
		⇕	⇕	⇕	⇕	⇕	(1-8)
[デルタ数列]	$\cdots$,	δ_{k+2},	δ_{k+1},	δ_k,	δ_{k-1},	δ_{k-2},	$\cdots$
		⇕	⇕	⇕	⇕	⇕	
（z 変換）	$\cdots$,	z^{+2},	z^{+1},	$z^0(=1)$,	z^{-1},	z^{-2},	$\cdots$

のように変数 z を対応付けして，

$$X(z)=\cdots+x[-2]z^{+2}+x[-1]z^{+1}+x[0]+x[1]z^{-1}+x[2]z^{-2}+\cdots$$
$$=\sum_{k=-\infty}^{\infty}x[k]z^{-k} \tag{1-9}$$

と表すことを考える（図1-5）．ここで，式(1-9)のディジタル信号全体の表現（べき級数の総和）は“z変換”と呼ばれる．

とくに，図1-3のようにデルタ数列 $\{\delta_k\}_{k=-\infty}^{k=\infty}$ を nT [秒]だけ右にずらした（遅らせた）信号 $\{x[k]\}_{k=-\infty}^{k=\infty}$ は，

$$x[k]=\delta_{k-n}=\begin{cases}1 & ;k=n\\ 0 & ;k\neq n\end{cases} \tag{1-10}$$

なので，式(1-9)に代入すると，

$$X(z)=z^{-n} \tag{1-11}$$

となることを考慮すれば，

$$nT\,[\text{秒}]\text{の遅れ} \quad \Leftrightarrow \quad z^{-n} \tag{1-12}$$

となる対応関係が成立することが分かる（図1-6，次ページ）．例えば，

「z^3」は，“$3T$[秒]だけ進んだサンプル時刻（$t=-3T$）”

「z^{-5}」は，“$5T$[秒]だけ遅れたサンプル時刻（$t=5T$）”

というように，変数 z のべき指数の符号に基づき，

プラス	（＋）	⇔	進ませる（左にずらす）
マイナス	（－）	⇔	遅らせる（右にずらす）

と読み解くのである．さらに，「Az^{-n}」の係数「A」は，

“$t=nT$ [秒]におけるディジタル信号値”

を表すことになる．つまり，

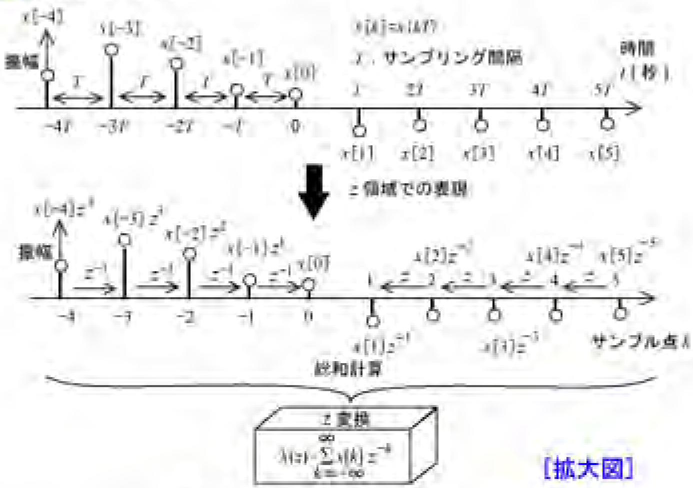
「$5z^{-3}$」

という表記は，

「$3T$[秒]における信号値は 5 である」

と翻訳し，ディジタル信号のサンプル時刻と信号値を同時に読み取る．

（実行例1-2，次ページ）



[拡大図]

図 1-5　ディジタル信号とZ変換

第二章

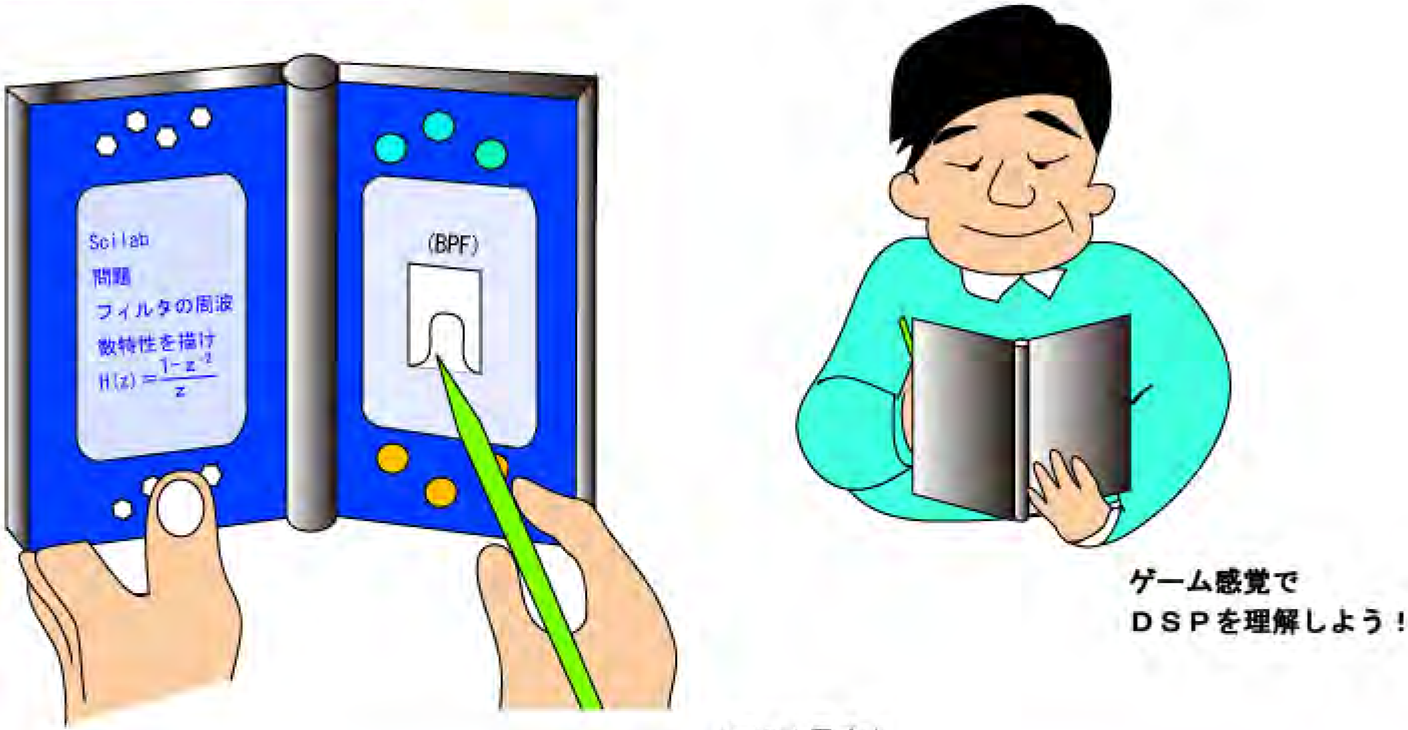
ディジタル·フィルタ解析の基礎をマスターしよう

前章では，ディジタル信号数学の気持ちをつかんでもらうために，信号処理の数式表現を物理的な言葉で“翻訳”するノウハウとともに，信号数学の真髄と言える「Z変換」にフォーカスして解説し終えたところである．

本章では，Z変換を土台にして，ディジタル·フィルタ解析の基礎を固めようというわけで，

差分方程式，伝達関数，インパルス応答，畳み込み処理，FIRフィルタ，IIRフィルタ

についての理解を深めながら，ディジタル·フィルタにおける「数式が織りなす華麗なる信号処理の世界」を楽しんでもらいたい（イラスト図2-1）．



イラスト図 2-1

二の２　ディジタル･フィルタの構成要素を差分方程式で表すと･･････【2/2】

④ メモリ（遅延器，シフトレジスタ）

1サンプリング間隔 T [秒]だけ数値を保持（記憶）しておく機能を有し，

$$y[k]=x[k-1] \tag{2-7}$$

と差分方程式で表される．この処理回路は，T [秒]だけ遅らせる処理（遅延器）に等価で，シフトレジスタで実現する（図2-4）．

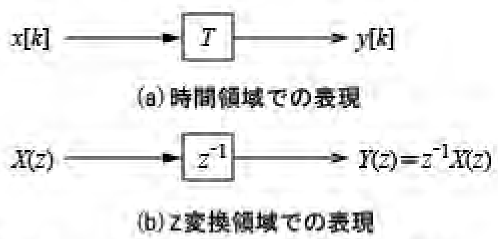


図 2-4　1個のメモリ（1サンプルの遅れ）

よって，mサンプル遅らせる処理は，

$$y[k]=x[k-m] \tag{2-8}$$

と差分方程式で表すことができ，そのブロック図は遅延回路を m個直列に接続すればよい（図2-5）．

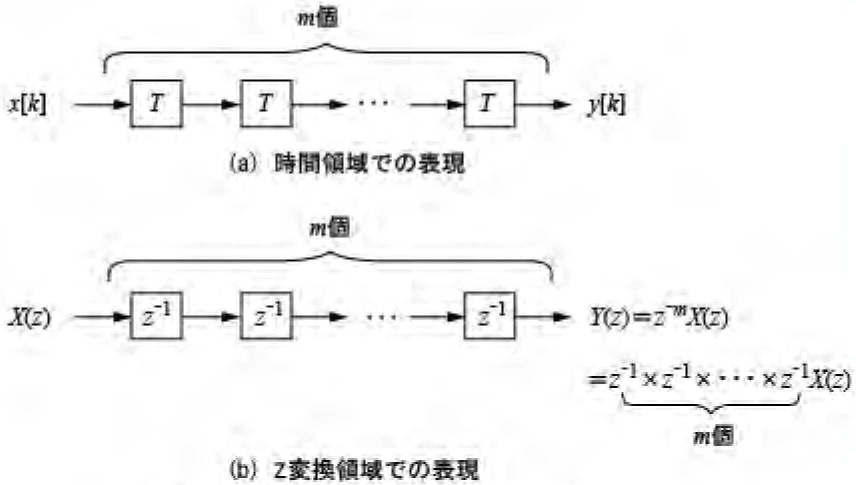


図 2-5　m個のメモリ（mサンプルの遅れ）

check BOX（正解と思う数字を半角で入力し“ENTER”キーを押す）

4個の隣り合う入力信号の平均値を出力するディジタル･フィルタの差分方程式は，

$$y[k]=(x[k]+x[k-1]+x[k-2]+x[k-3])\times 0.25$$

で与えられる．このとき，どのような周波数選択特性を有するか，実験室で確認せよ．

① ローパス　② ハイパス　③ バンドパス　　（答）

二の３　伝達関数は，入力信号の全体に対する出力信号の全体の比を表す

一般に，ディジタル·フィルタは，

『ディジタル化された入力信号系列 $\{x[k]\}_{k=-\infty}^{k=\infty}$ を

別の出力信号系列 $\{y[k]\}_{k=-\infty}^{k=\infty}$ に変換する機能を有する』

と考えることができる．したがって，ディジタル·フィルタにおける信号処理の中身を知るためには入出力の信号関係を調べる必要があり，ディジタル・フィルタの演算処理を表現するための基本式として差分方程式が重要となる．そこで，ディジタル信号全体を表す“z変換”を用いて，差分方程式から入出力の信号関係を導き出してみよう．

いま，ディジタル・フィルタの差分方程式を，

$$y[k]=0.5\,x[k]+0.5\,x[k-2] \qquad (2\text{-}9)$$

とし，$x[k]=y[k]=0$ $(k<0)$ とする．

まず，出力信号全体 $\{y[k]\}_{k=-\infty}^{k=\infty}$ を表すz変換 $Y(z)$ は，

$$Y(z)=y[0]+y[1]z^{-1}+y[2]z^{-2}+y[3]z^{-3}+\cdots$$

で定義されるので，式(2-9)を代入して，

$$Y(z)=\left\{0.5\,x[0]+\underbrace{0.5\,x[-2]}_{0}\right\}+\left\{0.5\,x[1]+\underbrace{0.5\,x[-1]}_{0}\right\}z^{-1}+\{0.5\,x[2]+0.5\,x[0]\}z^{-2}$$
$$+\{0.5\,x[3]+0.5\,x[1]\}z^{-3}+\cdots$$

となる．さらに続けて，式を整理すれば，

$$Y(z)=0.5\{x[0]+x[1]z^{-1}+x[2]z^{-2}+\cdots\}+0.5z^{-2}\{x[0]+x[1]z^{-1}+x[2]z^{-2}+x[3]z^{-3}+\cdots\}$$

であり，{　}の中が入力信号全体 $\{x[k]\}_{k=-\infty}^{k=\infty}$ のz変換 $X(z)$ に相当することから

$$Y(z)=(0.5+0.5z^{-2})\,X(z) \qquad (2\text{-}10)$$

と表される．ここで，

$$H(z)=0.5+0.5z^{-2} \qquad (2\text{-}11)$$

とおけば，入力と出力のz変換の間には，$H(z)$ を介して，

$$Y(z)=H(z)X(z) \qquad (2\text{-}12)$$

という関係が成り立つ(図2-6)．これは，z変換した世界でのディジタル・フィルタの別表現であり，この $H(z)$ は“伝達関数（あるいは，システム関数）”と呼ばれ，

出力信号の全体＝伝達関数×入力信号の全体　(2-13)

ということになる．

$$X(z) \longrightarrow \boxed{H(z)} \longrightarrow Y(z)$$

入力信号の全体　伝達関数　出力信号の全体

$$H(z)=\frac{\text{出力信号の全体}}{\text{入力信号の全体}}=\frac{\text{出力信号のz変換}}{\text{入力信号のz変換}}=\frac{Y(z)}{X(z)}$$

図 2-6　伝達関数の定義

check BOX（正解と思う数字を半角で入力し“ENTER”キーを押す）

いま，ディジタル・フィルタの差分方程式を，

$$y[k]=x[k]+x[k-1]+x[k-2]+\cdots+x[k-10]$$

とするとき，伝達関数を求めよ．ただし，$x[k]=y[k]=0$ $(k<0)$ とする．

① $\dfrac{1-z^{-10}}{1-z^{-1}}$　② $\dfrac{1-z^{-11}}{1-z^{-1}}$　③ $\dfrac{1-z^{-12}}{1-z^{-1}}$　（答）

その５　ＦＩＲフィルタの差分方程式と伝達関数

ＦＩＲ（Finite Impulse Response）フィルタは，インパルス応答の継続時間が有限なディジタル·フィルタである．つまり，入力信号が０になれば，その後インパルス応答の継続時間が過ぎたときに出力も０になる．

ところで，入出力関係を表すもっとも基本的な差分方程式は，

$$y[k]=a_0x[k]+a_1x[k-1]+a_2x[k-2]+\cdots+a_Mx[k-M] \qquad (2\text{-}19)$$

であり，よく使われる構成方法として直接形と転置形がある（図2-7，図2-8）．ここで，a_m（$m=0, 1, 2, \cdots, M$）はフィルタの係数と呼ばれるもので，Mはこのフィルタの次数である．

いま，直接形ＦＩＲフィルタにおいて，フィルタの（$M+1$）個の係数を下付き数字の順に a_0，a_1，a_2，$\cdots$，a_M と並べたものは，このフィルタのインパルス応答 $\{h[k]\}_{k=-\infty}^{k=\infty}$ すなわち，

$$h[k]=\begin{cases} 0 & ;k<0 \\ a_k & ;0\leqq k\leqq M \\ 0 & ;M<k \end{cases} \qquad (2\text{-}20)$$

に一致する．

これに対して，図2-8に示す転置形ＦＩＲフィルタの差分方程式は，

$$\begin{cases} w_M[k]=a_Mx[k] \\ w_{M-1}[k]=a_{M-1}x[k]+w_M[k-1] \\ w_{M-2}[k]=a_{M-2}x[k]+w_{M-1}[k-1] \\ \quad\vdots \\ w_1[k]=a_1x[k]+w_2[k-1] \\ w_0[k]=a_0x[k]+w_1[k-1] \\ y[k]=w_0[k] \end{cases} \qquad (2\text{-}21)$$

と表される連立差分方程式になる．このとき，フィルタの係数を直接形の逆順 a_M，a_{M-1}，a_{M-2}，$\cdots$，a_0 と並べたものが，式(2-20)のインパルス応答になる．

いずれのＦＩＲフィルタも，式(2-19)と式(2-21)に対する伝達関数は，

$$H(z)=a_0+a_1z^{-1}+a_2z^{-2}\cdots+a_Mz^{-M} \qquad (2\text{-}22)$$

で表されるので，各自で検証してもらいたい（付録Ｊ）．

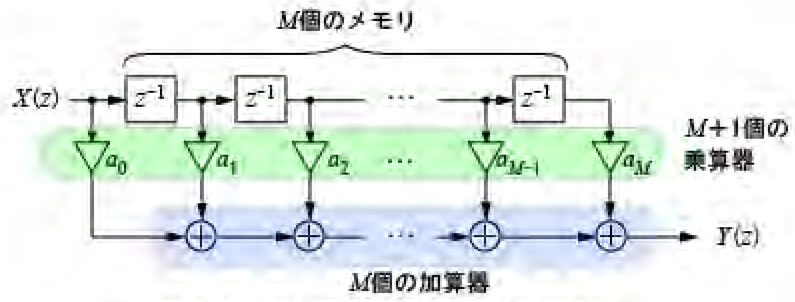


図 2-7　M次のFIRフィルタ（直接形構成）

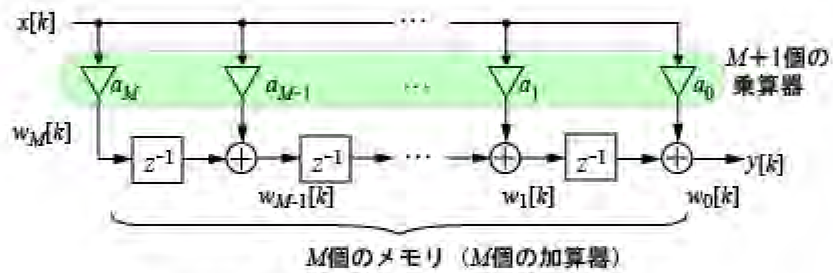


図 2-8　M次のFIRフィルタ（転置形構成）

二の６

ＩＩＲフィルタの差分方程式と伝達関数【1/3】

ＩＩＲ(Infinite Impulse Response)フィルタは，インパルス応答の継続時間が無限なディジタル・フィルタである．このフィルタでは，たとえ入力信号が０になったとしても，出力０にはならない．ＩＩＲフィルタの入出力関係を表すもっとも基本的な差分方程式は，

$$y[k]=b_1y[k-1]+b_2y[k-2]+\cdots+b_Ny[k-N]+a_0x[k]+a_1x[k-1]+a_2x[k-2]+\cdots+a_Mx[k-M] \qquad (2\text{-}23)$$

となる．このとき，伝達関数は，

$$H(z)=\frac{a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_Mz^{-M}}{1-b_1z^{-1}-b_2z^{-2}-\cdots-b_Nz^{-N}}=\frac{A(z)}{B(z)} \qquad (2\text{-}24)$$

で与えられるので，各自で検証してもらいたい．なお，式(2-23)と式(2-24)において，分母の最高次数Nと分子の最高次数Mは異なる値でかまわないが，以下では$N=M$であると仮定して話を進める（$N\neq M$の場合は，いくつかの係数が０であるとみなせばよいので$N=M$としても一般性は損なわれない）．

ＩＩＲフィルタは，過去の出力がフィードバック（帰還）されるので，安定性に注意する必要がある（姉妹書「わかるディジタル信号処理～基礎編」三の10を参照）．

式(2-23)で示す差分方程式をそのまま構成すると図2-10となり，直接形Ｉと呼ばれている．直接形Ｉでは，点線で囲んだ二つのブロックが直列接続されたものとみなすことができ，左側のブロック関数は，

$$A(z)=a_0+a_1z^{-1}+a_2z^{-2}+\cdots+a_Mz^{-M} \qquad (2\text{-}25)$$

で，右側のブロックの伝達関数は，

$$\frac{1}{B(z)}=\frac{1}{1-b_1z^{-1}-b_2z^{-2}-\cdots-b_Nz^{-N}} \qquad (2\text{-}26)$$

である．全体の伝達関数は式(2-25)と式(2-26)の二つの伝達関数の積となり，式(2-24)が得られる．

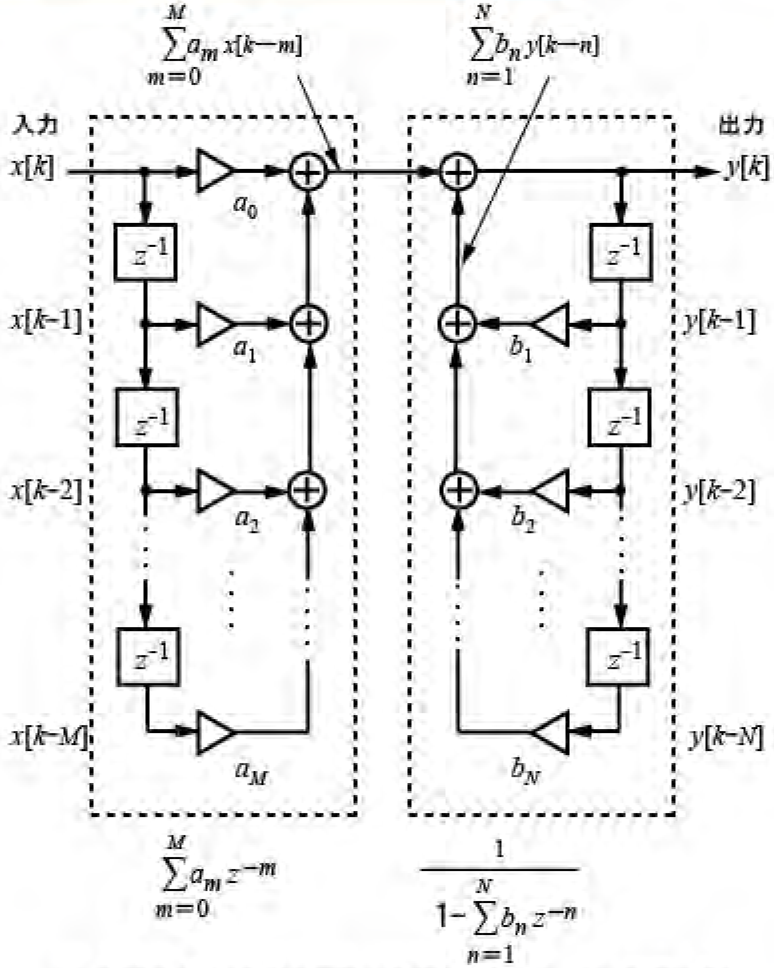


図 2-10 IIRフィルタのブロック図（直接形Ｉ，Ｍ＝Ｎの場合）

二の６　ＩＩＲフィルタの差分方程式と伝達関数【3/3】

それでは，直接形Ｉ（上段）と標準形（下段）が同じ入出力応答ならびに周波数特性をもつことを，図2-12の二つのフィルタ構成における係数値を同じ値に設定して確かめてみよう．

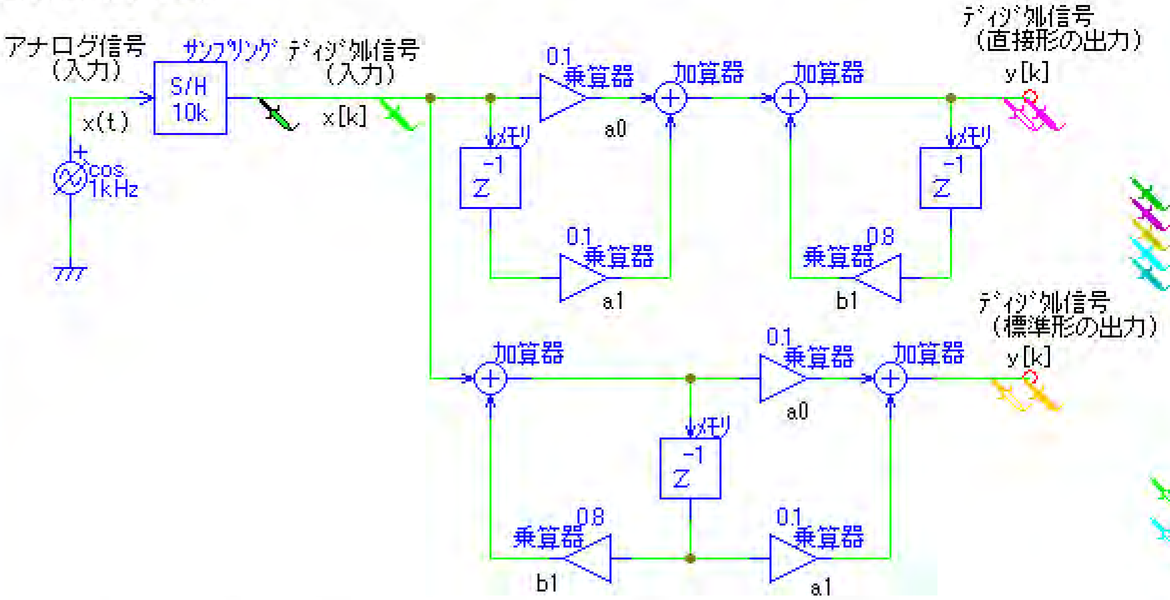
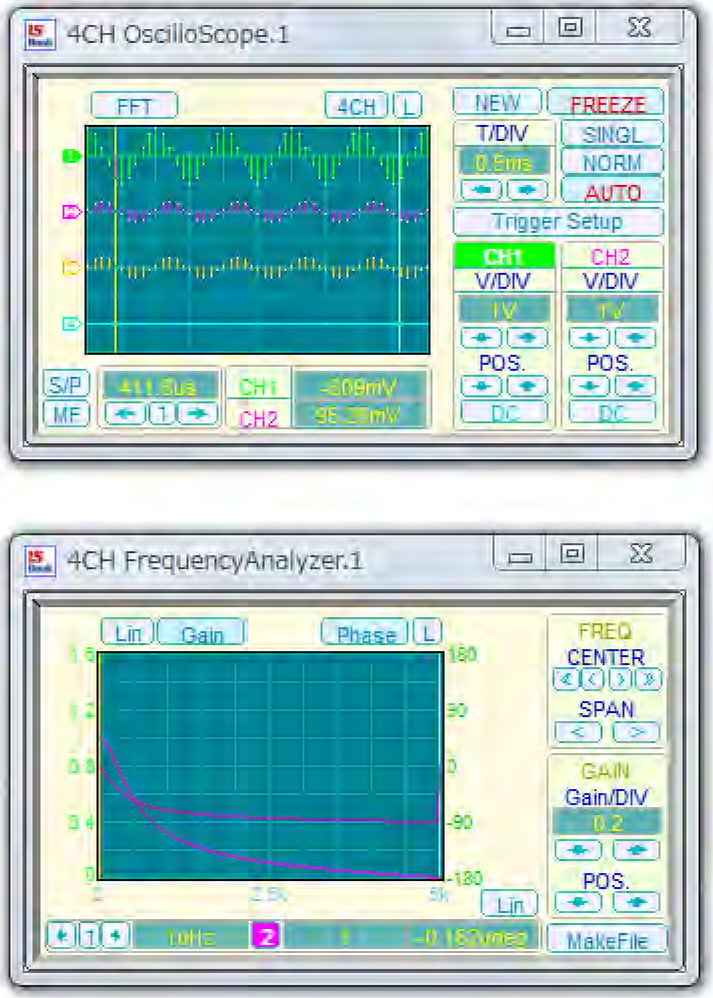


図 2-12　メモリの個数を少なくしても同じ特性になる！



第三章　ディジタル･フィルタの周波数特性を理解しよう

ディジタル･フィルタは，ある周波数成分を取捨選択したい場面で大活躍する信号処理を実現するものである．ディジタル信号（数値データ）からある周波数成分を抽出したり，雑音を除去したりするために用いられ，周波数選択（フィルタリング；filterling）特性を有する．

また，ディジタル･フィルタは単に四則計算するだけの演算処理システムであるにも関わらず，周波数の高低によって出力の大きさをコントロールができる．何とも不思議な感じのするディジタル･フィルタの周波数特性の定義，計算の仕方に習熟しておくことは，ディジタル信号処理を知るうえでの最大のポイントと言えるものなので，じっくりと読み進めてもらいたい．

まず，ものは試しということで，Scilab画面を開いて，いろいろな伝達関数を入力し，周波数特性やインパルス応答をグラフ表示してみよう．

使用例　伝達関数 $H(z)=\dfrac{1+z^{-1}}{2}$

Scilabでの書き方の例を示す．ただし，伝達関数 $H(z)$ は，変数 z^{-1} を変数 p で置き換えて入力する．

実行例3-a

```
-->ztspec((1+p)/2,1);
```

実行例3-b

```
-->hz=(1+p)/2;
-->ztspec(hz,1);
```

実行結果として，グラフ･ウィンドウ画面上に三つのグラフ，すなわちインパルス応答（上段で赤色），ゲイン（中段で緑色）と位相（下段でピンク色）の周波数特性が表示される（実行図3-1）．

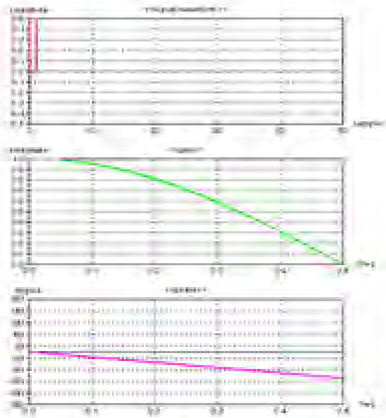
いくつかの伝達関数 $H(z)$ を以下に示しておくので，フィルタ特性をグラフ表示してみてほしい．

① $\dfrac{1+z^{-1}+z^{-2}+z^{-3}+z^{-4}}{5}$　② $0.2\times\dfrac{1-z^{-5}}{1-z^{-1}}$　③ $0.25\times\dfrac{1+z^{-2}}{1-0.5z^{-2}}$

④ $0.25\times\dfrac{1+2z^{-1}+z^{-2}}{1-0.5z^{-1}+0.5z^{-2}}$　⑤ $0.25\times\dfrac{1-z^{-2}}{1-0.5z^{-1}+0.5z^{-2}}$

⑥ $\dfrac{0.5-0.5z^{-1}+z^{-2}}{1-0.5z^{-1}+0.5z^{-2}}$　⑦ $0.5\times\dfrac{1-2z^{-1}+z^{-2}}{1-0.5z^{-1}+0.5z^{-2}}$

なお，①と②はローパスで同一特性，③はバンド･エリミネート，④はローパス，⑤はバンドパス，⑥はオールパス，⑦はハイパスとなるので，落ち着いてScilabの処理命令を入力してフィルタ特性をチェックしてください．



実行図3-1

三の２　周波数（選択）特性は極座標表示でわかる！【2/2】

◆ 偏角 θ は位相（フェーズ，phase）

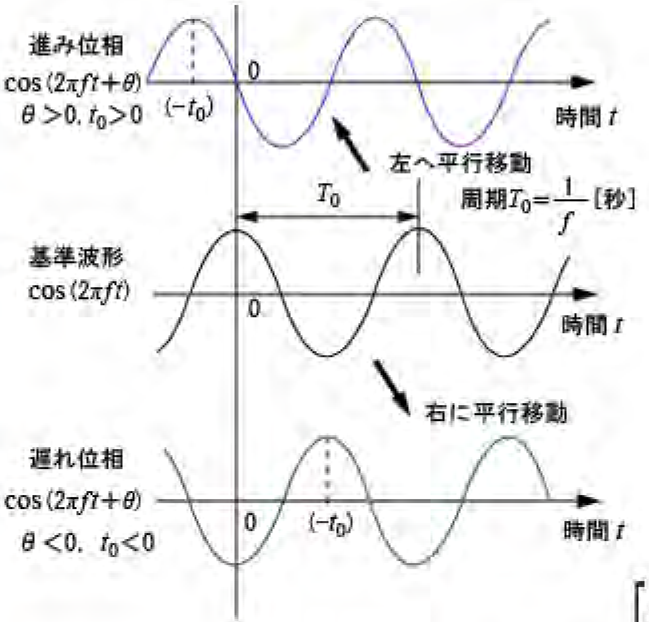
伝達特性 $G(f)$ の偏角，すなわち，

$$\angle G(f)=\angle Y(f)-\angle X(f) \qquad (3\text{-}12)$$

は入力と出力の偏角との差で，時間ずれ t を角度に換算した値を表す．一般的には，入力信号の周波数 f [Hz]，時間ずれを t_0 [秒] とするとき，位相 $\angle G(f)$ は，

$$\angle G(f)=2\pi f t_0 \quad （あるいは\ \omega t_0） \qquad (3\text{-}13)$$

となる．ここで，位相が正（プラス）のときは進み位相と呼んで入力信号を左に平行移動したものが出力され，負（マイナス）のときは遅れ位相といって右に平行移動したものが出力される(図3-5)．



位相 θ [rad]	時間ずれ t_0[秒]	
$-\pi$	$-T_0/2$	遅れ位相
$-\pi/2$	$-T_0/4$	遅れ位相
0	0	同相
$\pi/2$	$T_0/4$	進み位相
π	$T_0/2$	進み位相

$$\left[\theta=2\pi f t_0 \ （あるいは，t_0=\frac{\theta}{2\pi f}）\right]$$

図 3-5　位相と時間ずれの相互関係

また，位相 $\angle G(f)$ [rad]から時間ずれ t_0 [秒]は，式(3-13)より，

$$t_0=\frac{位相}{角周波数}=\frac{\angle G(f)}{2\pi f}=\frac{\angle G(f)}{\omega} \qquad (3\text{-}14)$$

と算出できる．

以上のことをまとめると，フィルタへの入力 $x(t)$ として，

$$x(t)=e^{j2\pi ft} \qquad (3\text{-}15)$$

で表される周波数 f [Hz]の複素正弦波を加えたとき，出力信号 $y(t)$ が，

$$y(t)=Ae^{j2\pi f(t+t_0)} \qquad (3\text{-}16)$$

であったとする．つまり，出力振幅が入力振幅の A 倍，時間ずれが t_0 [秒]であると翻訳できるというわけだ．なお，時間ずれが負であれば“遅れ”，正であれば“進み”ということになる．

さらに，式(3-15)と式(3-16)の入出力の比をとれば，

$$\frac{y(t)}{x(t)}=\frac{Ae^{j2\pi f(t+t_0)}}{e^{j2\pi ft}}=Ae^{j2\pi ft_0}=Ae^{j\omega t_0}=G(f) \qquad (3\text{-}17)$$

となる関係が得られ，周波数に対するシステムの性質を記述したものである．

三の３　四則計算で周波数（選択）特性をコントロールできる？

ところで，ディジタル・フィルタでは四則計算だけで，コイルやコンデンサなどの周波数に依存する電気的な回路素子は何もないのに，周波数選択特性がある．「どうしてなんだろう．なぜ？」という素朴な疑問を抱かれるとしても何の不思議もない．

それでは，3種類の構成要素（メモリ，乗算，加算）のすべてを含むディジタル・フィルタ（図3-6）を考え，周波数選択特性を調べてみよう．図3-6 のディジタル・フィルタにおける信号処理演算は，

$$y[k]=bx[k]+x[k-1] \tag{3-18}$$

で表される差分方程式で記述される．

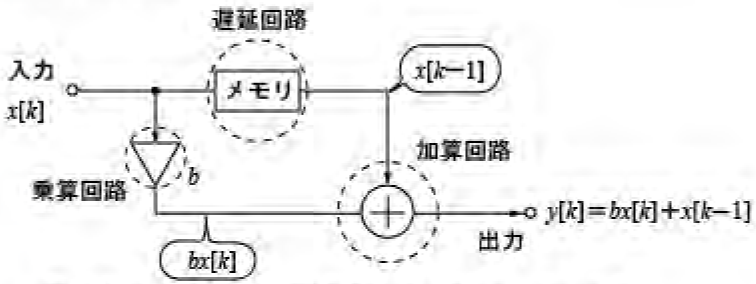


図 3-6　3種類の演算回路を含むディジタル・フィルタ

周波数特性を解析するには，式(3-15)で示す複素正弦波 $(x(t)=e^{j2\pi ft})$ を入力したときの出力信号を求めればよかったわけですね．そこでまず，入力するディジタル信号を，式(3-15)においてサンプリング処理する．すなわち $t=kT$（T[秒]はサンプリング間隔）を代入することにより，

$$x[k]=x(kT)=e^{j2\pi fkT} \tag{3-19}$$

と表される．式(3-19)より，

$$x[k-1]=e^{j2\pi f(k-1)T} \tag{3-20}$$

であり，式(3-18)の差分方程式に代入すれば，

$$y[k]=be^{j2\pi fkT}+e^{j2\pi f(k-1)T}=b^{j2\pi fkT}+e^{j2\pi fkT}\times e^{-j2\pi fT}$$
$$=e^{j2\pi fkT}(b+e^{-j2\pi fT})$$

となり，さらに式(3-19)より，

$$y[k]=x[k](b+e^{-j2\pi fT})$$

と変形され，最終的に，

$$\frac{y[k]}{x[k]}=b+e^{-j2\pi fT} \tag{3-21}$$

と表されるディジタル・フィルタの周波数特性の式が導き出せる．

次に，式(3-21)をじっくり見てみると，周波数 f が含まれている項 $e^{-j2\pi fT}$ は，式(3-18)から周波数特性の式(3-21)が導き出されるプロセスから $x[k-1]$ に相当していることがわかる．もう，おわかりのことだと思うが，

ディジタル・フィルタの周波数の素　＝　メモリ

と言えるのである（図3-7）．つまり，一時的に数値を保持するメモリ（遅延）を含むディジタル・フィルタの四則計算によって，周波数選択特性をコントロールできることがわかる．極言すれば，アナログ・フィルタのコイルやコンデンサなどの回路素子を四則計算で作れるというわけだ（感動ものですね）．

図 3-7　ディジタル・フィルタの周波数の素

三の６

伝達関数からわかる周波数特性【2/2】

実行例3-3では，サンプリング時間 $T=1/120$[秒]でサンプリング周波数 $f_S=120$[Hz]なので，$0\leq f<60$[Hz]の周波数範囲において，

$$H(z)=\frac{1-3z^{-1}+3z^{-2}-z^{-3}}{8}=0.125-0.375z^{-1}+0.375z^{-2}-0.125z^{-3} \qquad (3\text{-}33)$$

で表される伝達関数 $H(z)$ のゲインと位相の周波数特性をグラフ表示する（図3-12）．ここで，ゲインと位相の理論値はそれぞれ，

$$\text{ゲイン特性：}\left|H\left(e^{j\frac{\pi f}{60}}\right)\right|=\frac{1}{8}\left|\sin\left(\frac{\pi f}{40}\right)-3\sin\left(\frac{\pi f}{120}\right)\right| \qquad (3\text{-}34)$$

$$\text{位相特性：}\angle H\left(e^{j\frac{\pi f}{60}}\right)=\begin{cases}-\dfrac{\pi}{2}-\dfrac{\pi f}{40} & ：0\leq f<20\\[2ex] \dfrac{3\pi}{2}-\dfrac{\pi f}{40} & ：20\leq f<60\end{cases} \qquad (3\text{-}35)$$

で与えられる（付録H を参照）．

実行結果からは，ハイパス特性を有する直線位相のディジタル･フィルタであることがわかる．また，式(3-33)を有するディジタル･フィルタは図3-12のように構成でき，ＤＳＰシミュレータ（周波数アナライザ）による周波数特性は Scilabで伝達関数から求めた結果に一致することが確かめられる．なお，位相特性における360度のジャンプが見られるが，理由は位相が（-180）度より遅れた分を便宜上（+180）度から描いていることによる．

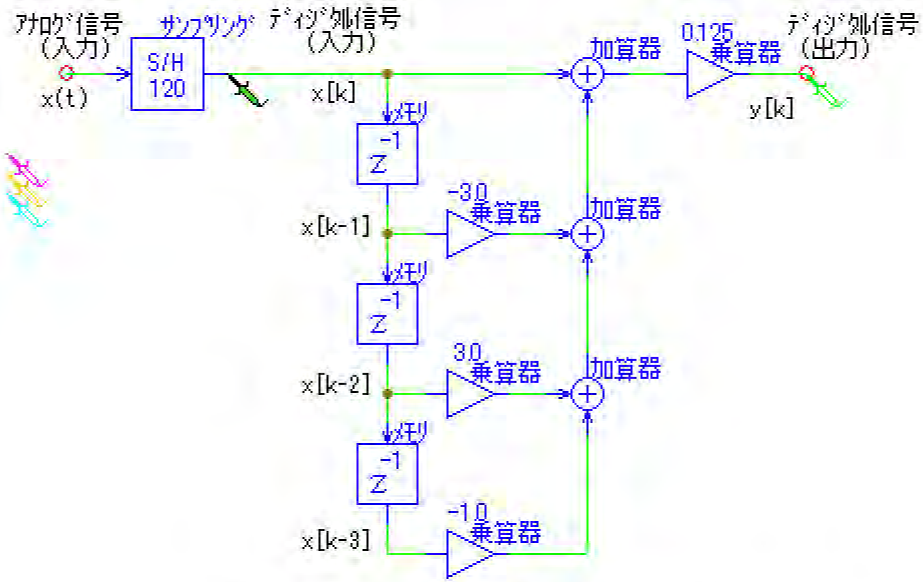
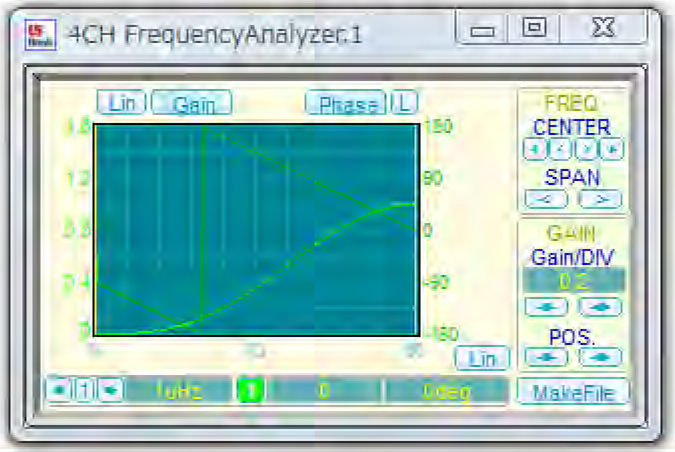


図 3-12　式(3-33)の伝達関数を有するFIRフィルタ



check BOX（正解と思う数字を半角で入力し"ENTER"キーを押す）

次の伝達特性 $H(z)$ を有するディジタル・フィルタのゲイン特性はどれか．

$$H(z)=\frac{1-3z^{-2}+3z^{-4}-z^{-6}}{8}$$

① ローパス　② ハイパス　③ バンドパス　④ バンド・エリミネート

（答）

三の７　インパルス応答からわかる周波数特性【2/2】

図3-14より，インパルス応答系列と伝達関数から求めた周波数特性が一致するので，ディジタル・フィルタ設計においてアナログ・フィルタのインパルス応答に一致させるように伝達関数を近似する手法が提案されている．例えば，アナログＲＣ回路とＩＩＲフィルタのインパルス応答を一致させると，ほぼ同一の周波数特性を実現できます（図3-15）．cos波入力の周波数を変えて，アナログとディジタルの入出力信号の変化，ローパス特性を有することを確認してもらいたい．

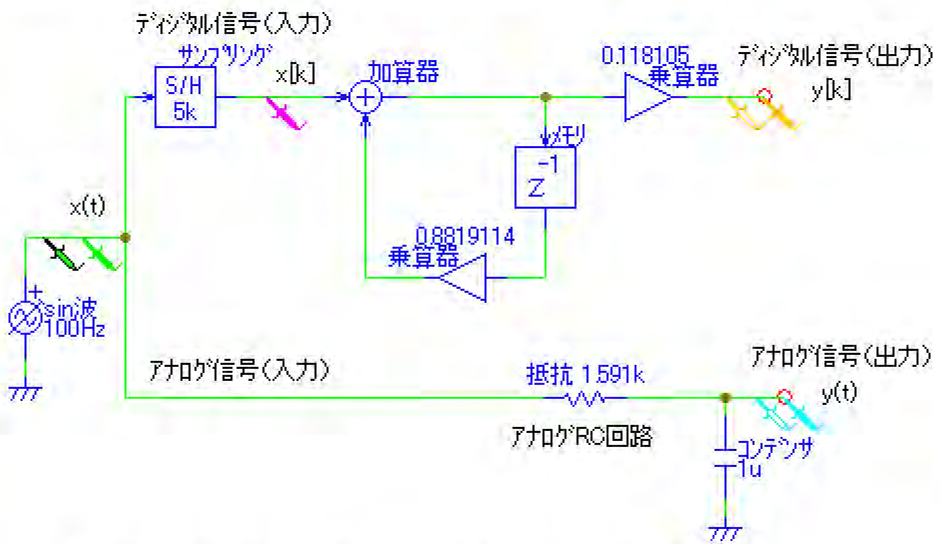
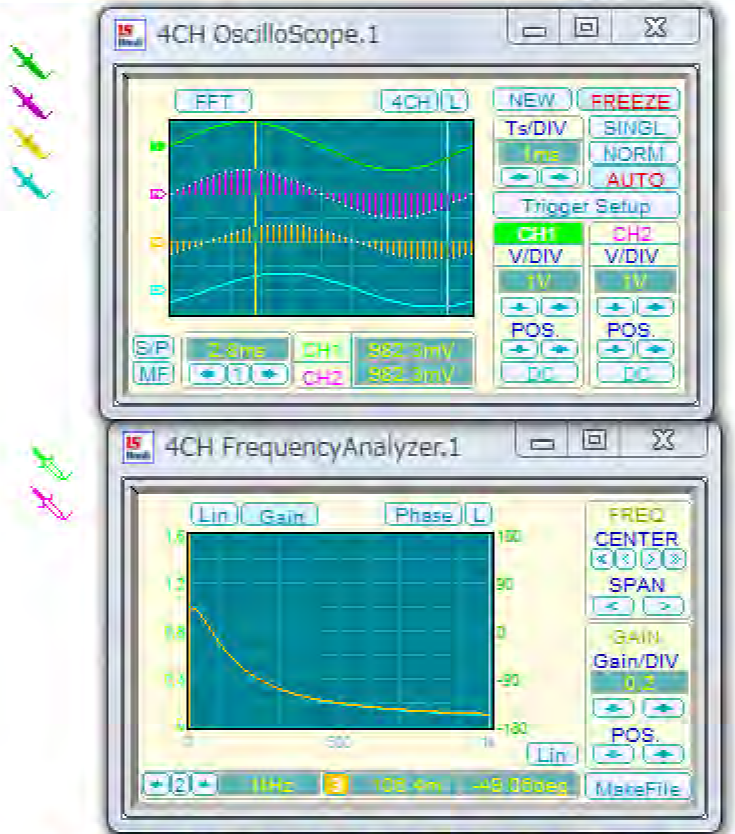


図 3-15　アナログＲＣ回路をディジタル・フィルタで実現する



check BOX（正解と思う数字を半角で入力し“ENTER”キーを押す）

いま，インパルス応答 $\{h[k]\}_{k=-\infty}^{k=\infty}$ が

$$h[k]=\begin{cases}0 & : k<0\\ 1.1^{-k} & : k\geqq 0\end{cases}$$

となるディジタル・フィルタの周波数特性はどれか．

①ローパス　②ハイパス　③バンドパス　　（答）

第四章　ディジタル・フィルタをハード＆ソフトで作ってみよう【1/2】

前章までは，ディジタル・フィルタの表現方法として差分方程式，ブロック図(ハードウェア構成)，伝達関数，零点と極などの数式的な取り扱いをメインに解説してきた．しかしながら，数式表現だけでは机上の空論になりかねないので，本章からはディジタル・フィルタをＤＳＰアナライザでハードウェア構成したり，ソフトウェア(ScilabによるＤＳＰプログラム)で実現して，ディジタル信号処理のようすを体感してもらうことにする．

簡単な例(ランダム雑音を取り除く１次のＩＩＲフィルタ)を示そう．

いま，差分方程式が，

$$y[k]=b_1y[k-1]+a_0x[k]+a_1x[k-1]\ ;\ b_1=0.8\ ,\ a_0=0.1\ ,\ a_1=0.1 \qquad (4\text{-}a)$$

で表されるとき，図4-0のようにハードウェア構成できる．また，伝達関数 $H(z)$ は，式(4-a)の両辺をz変換して算出すれば，

$$H(z)=\frac{Y(z)}{X(z)}=\frac{0.1+0.1z^{-1}}{1-0.8z^{-1}}\left(=\frac{a_0+a_1z^{-1}}{1-b_1z^{-1}}\right) \qquad (4\text{-}b)$$

となる．ここで，分母多項式 z^{-1} の係数 $(-b_1)$ の符号を反転したものが，差分方程式の $y[k-1]$ の係数(ハードウェア構成の乗算器b1に相当)になることに注意してもらいたい．

図4-0のオシロスコープ表示を見ると，雑音を含む入力信号(緑色)の雑音が取り除かれた出力信号(ピンク色)が得られることがわかる．同時に，周波数アナライザ表示からは，ローパス特性を有することも確認できる．また，cos波の周波数や雑音の大きさを変えて，ＩＩＲフィルタによる信号処理のようすを体験してもらいたい．

また，式(4-a)の差分方程式に基づき，

$$\begin{cases} w[k]=b_1w[k-1]+x[k] & (4\text{-}c) \\ y[k]=a_0w[k]+a_1w[k-1] & (4\text{-}d) \end{cases}$$

と変形し，これらの計算をソフトウェア実現したものとしてScilabによるＤＳＰプログラムを作成する(プログラム例4-0)．ＤＳＰプログラミングについては，付録Ｆ と 付録Ｇ を参考にしながら実際にプログラム作成・実行を体験しておこう．

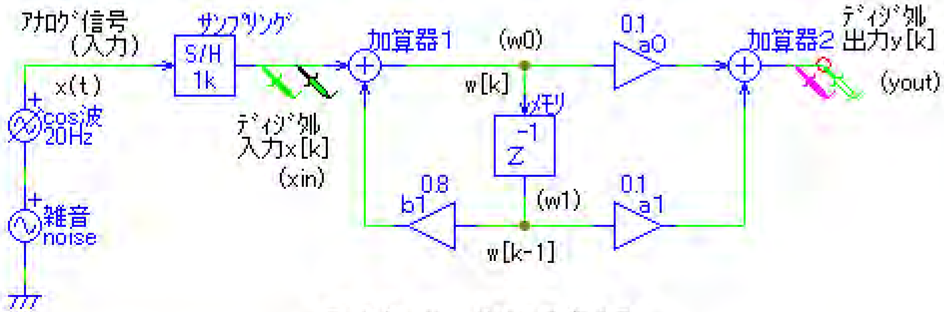
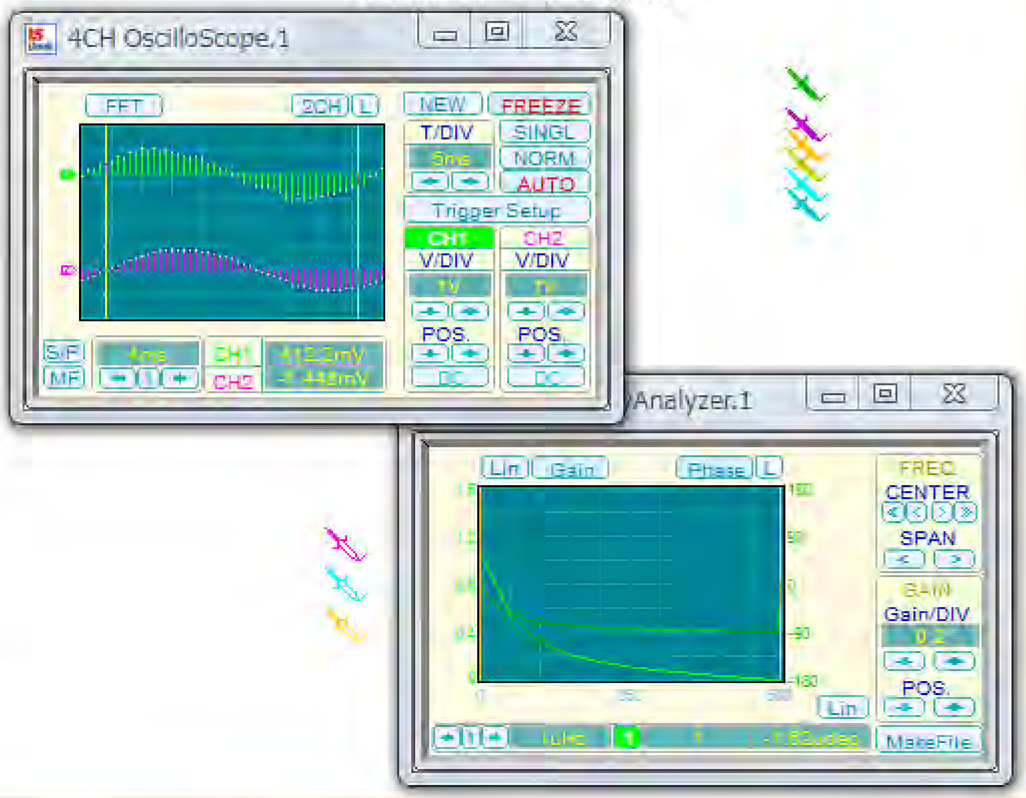


図 4-0　ハードウェア構成例



四の１　「帰還なし（ＦＩＲフィルタ）」の差分方程式から作ると・・・・・【1/2】

いま，ＦＩＲフィルタの入出力関係を表す差分方程式が，

$$y[k]=0.25x[k]+0.5x[k-1]+0.25x[k-2] \qquad (4\text{-}1)$$

で表されるとき，ハードとソフト（ScilabによるＤＳＰプログラム）で作ってみよう．図4-1には，２個のメモリと２個の２入力１出力の加算器，２個の乗算器が準備されているので，部品を移動･接続してハードウェア構成を完成させてもらいたい【正解はIS Bookの[ユーザーサポートページ(要ユーザー登録)]に掲載】．

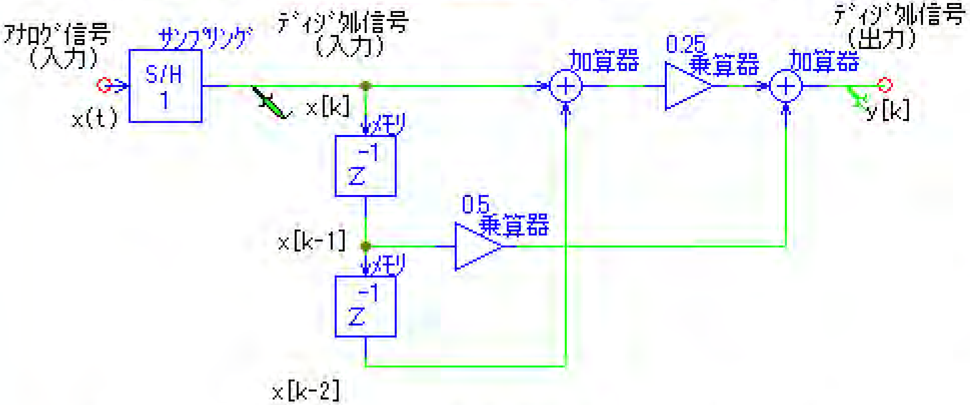
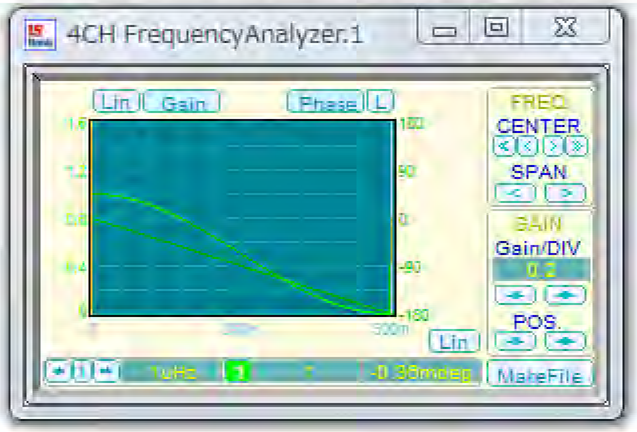


図 4-1　式(4-1)の差分方程式で表されるFIRフィルタ（帰還なし）



式(4-1)より，３個の乗算器が必要と思われるかもしれない．でも，２個しか用意されていないので，

$$y[k]=0.25\,[x[k]+x[k-2]]\,+0.5x[k-1] \qquad (4\text{-}2)$$

と変形すれば実現できそうである．是非とも，正解を見つけ出してほしい．完成した人は，周波数アナライザを利用してゲインと位相の各特性を調べて，ゲインはローパス特性，位相は周波数に正比例することを確認しておこう．

次は，Scilabを用いてＤＳＰプログラミングに挑戦！

最初のステップは，式（4-2）の変数の割り当てからで，ここでは，

- $x[k]$ ⇔ xin　（入力信号の変数名 xinの変更は不可）
- $x[k-1]$ ⇔ x1
- $x[k-2]$ ⇔ x2
- $y[k]$ ⇔ yout　（出力信号の変数名 youtの変更は不可）

で表すことにする．付録Ｆ を参考に，ＦＩＲフィルタの基本プログラムのアミカケ部分だけを図4-1のハード構成，あるいは式(4-2)に合わせて，プログラム例4-1 に示す①，②，③を書き直す．続いて，ファイル名(例えば，prog41.sce)を付けてフォルダ DspViewer に保存し，実行すればよい(実行例4-1，図4-2)．

[プログラム例4-1]（式(4-2)のＤＳＰプログラム）

```
function[youtput] = FILTER(xinput)
//
//****  initialize(shift register, memory)****
  x1=0; x2=0;                                         ･････････①
//*********************************************
//
   for k = 1:1:length(xinput)
         xin = xinput(k);
//
//*******  output ,shifting *****************
    yout = 0.25*(xin + x2) + 0.5*x1;                  ･････････②
    x2=x1; x1=xin;                                    ･････････③
//*********************************************
      youtput(k) = yout;
   end;
endfunction
```

四の３　多項式で表される伝達関数から作ると・・・・・【1/3】

いま，伝達関数（ＦＩＲフィルタ）が，

$$H(z)=0.125\times(1+z^{-1})^3 \tag{4-6}$$

で表されるとき，ハードとソフトで作ってみよう．ここで，式(4-6)は，

$$H(z)=0.125+0.375z^{-1}+0.375z^{-2}+0.125z^{-3} \tag{4-7}$$

と多項式として展開できるが，入力と出力の信号が見えていない．でも，伝達関数が入力信号全体に対する出力信号全体の比を表すことを思い出してもらえると，

$$H(z)=\frac{Y(z)}{X(z)}=0.125+0.375z^{-1}+0.375z^{-2}+0.125z^{-3} \tag{4-8}$$

となる関係より，

$$Y(z)=H(z)X(z)=0.125X(z)+0.375z^{-1}X(z)+0.375z^{-2}X(z)+0.125z^{-3}X(z) \tag{4-9}$$

が得られる（二の３ を参照）．式(4-9)において，$z^{-m}X(z)$は mサンプル前の入力信号，すなわち $x[k-m]$であり，m個のメモリを直列接続して実現できる．

さて，図4-5には，３個のメモリと３個の入力１出力の加算器，２個の乗算器が準備されているので，部品を移動･接続してハードウェア構成を完成させてもらいたい．

式(4-9)より，３個の乗算器が必要と思われるかもしれない．でも，２個しか用意されていないので，

$$Y(z)=0.125\times\{X(z)+z^{-3}X(z)\}+0.375\times\{z^{-1}X(z)+z^{-2}X(z)\} \tag{4-10}$$

と変形すれば実現できそうである．是非とも，正解を見つけ出してほしい．完成した人は，周波数アナライザを利用してゲインと位相の各特性を調べて，ゲインはローパス特性，位相は周波数に正比例することを確認しておこう【正解はIS Bookの[ユーザーサポートページ(要ユーザー登録)]に掲載】．

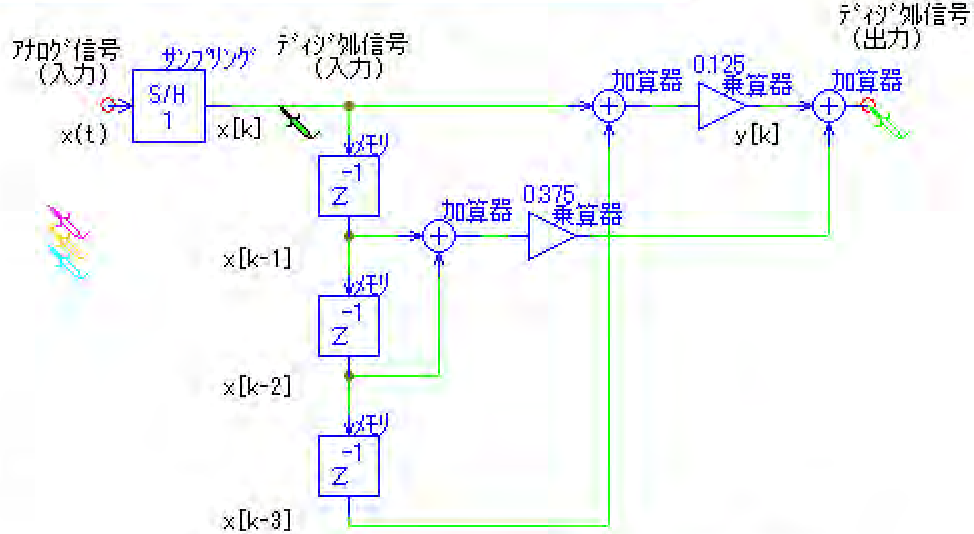
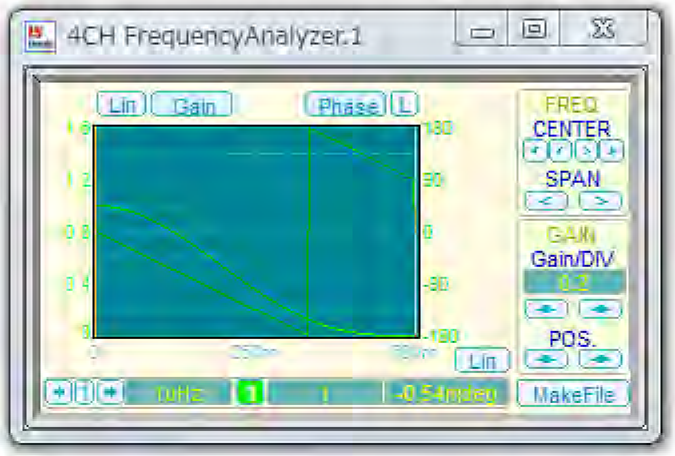


図 4-5　多項式で表される伝達関数のFIRフィルタ[(式4-6)]



四の５　伝達関数の積から作ると‥‥‥【1/2】

いま，伝達関数（ＩＩＲフィルタ）が，

$$H(z)=\frac{0.2\times(1+z^{-3})}{1-0.6z^{-3}} \tag{4-16}$$

で表されるとき，因数分解して積形式で表してみよう（実行例4-6）．

実行例4-6　（伝達関数を入力して因数分解する）

```
—>hz = 0.2*(1+p^3)/(1-0.6*p^3);   ……①
—>fact(hz)                         ……②
                    2
       1 - p + p                   1 + p           0.2
  ---------------------------   --------------    -----
                          2
  1 + 0.8434327p + 0.7113787p   1 - 0.8434327p      1
```

［実行例4-6 の説明］

① z^{-1}をプログラム変数 p で置き換えて，式(4-16)を入力する

② 関数コマンド factは，フィルタの伝達関数を因数分解する．この例で得られた結果から，次のように表される．

$$H(z)=0.2\times\frac{1-z^{-1}+z^{-2}}{1+0.8434327z^{-1}+0.7113787z^{-2}}\times\frac{1+z^{-1}}{1-0.8434327z^{-1}} \tag{4-17}$$

それでは，式(4-17)に基づき，ハードとソフトで作ってみよう．作る前に，伝達関数を簡略化して，

$$H(z)=H_0\times H_1(z)\times H_2(z) \tag{4-18}$$

ただし，

$$H_0=0.2,\ H_1(z)=\frac{1-z^{-1}+z^{-2}}{1+0.8434327z^{-1}+0.7113787z^{-2}}\ ,\ H_2(z)=\frac{1+z^{-1}}{1-0.8434327z^{-1}}$$

と表すことにする．このとき，伝達関数の定義から，

$$H(z)=H_0\times H_1(z)\times H_2(z)=\frac{Y(z)}{X(z)}$$

なので，

$$P(z)=H_0H_1(z)\ ,\ Q(z)=H_1(z)P(z),\ Y(z)=H_2(z)X(z) \tag{4-19}$$

と表して，三つの伝達関数を順に直列接続すれば実現できることがわかる（図4-10，縦続形構成という）．

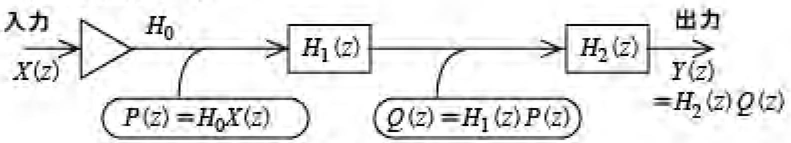


図4-10　縦続形構成

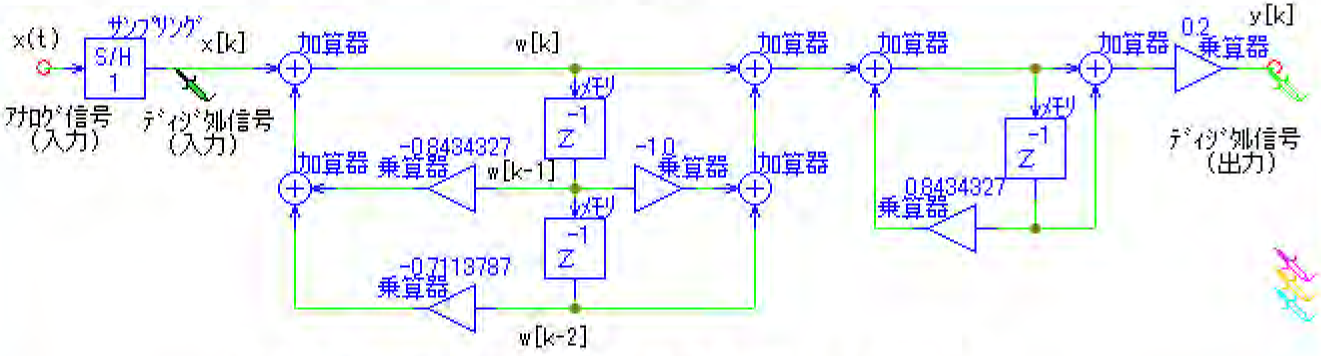


図 4-11 式(4-17)の伝達関数の積（縦続形構成）で表されるIIRフィルタ