験震時報第56巻
(1993) 17〜34頁

ベッセルディジタルフィルタの自動設計について

勝間田明男*

Automatic Designing of Bessel Digital Filters

Akio KATSUMATA*

(Received March 3, 1992)

§1. はじめに

数値的に記録された地震波形データを処理・解析する場合にディジタルフィルタは必要不可欠なものとなっている。ディジタルフィルタには，FIR（Finite Impulse Response）型とIIR（Infinite Impulse Response）型がある（Hamming，宮川・今井訳，1980)。代表的FIR型フィルタでは次のようにして出力を得る。

$$y_i = \sum_{k=-N}^{N} a_k x_{i-k}$$

ここで x_i は入力，y_i は出力，a_k はフィルタの係数，Nは定数である。IIR型フィルタは次のようにして出力を得る。

$$y_i = \sum_{k=0}^{N} a_k x_{i-k} + \sum_{k=1}^{N} b_k y_{i-k}$$

ここで x_i は入力，y_i は出力，a_k，b_k はフィルタの係数，N，Mは定数である。IIRフィルタは，ある時点の出力を得る場合にそれ以前の出力データを用いて出力を計算するために，無限に続くインパルス応答を持つ。IIRフィルタの設計には，電気回路フィルタの設計理論が適用され，効率的に必要とする特性が得られるようになっている。

斎藤（1978）によりIIRフィルタの設計理論とプログラムが紹介され，それが地震関係者に広く利用されている。斎藤によるものは，バタワース型及びチェビシェフ型等のフィルタである。ベッセル型（遅延平坦特性）のディジタルフィルタについて，次数に制限のない設計手法を開発した。ベッセルフィルタは，遅延時間の周波数依存性が小さいことが特徴となっている。この種のフィルタを必要とする関係者の便宜も考慮し，ここに紹介する。

§2. 理論

電気回路の分野で，特定の周波数成分を取り出す手段として様々なフィルタが考案されてきた。代表的な周波数特性としてバタワース型やチェビシェフ型などが知られている。その設計理論は，IIR型のディジタルフィルタの周波数特性の設計にも適用されている。バタワース型は周波数特性が次のような関数となるものである。

$$|F(x)| = \frac{1}{\sqrt{1 + x^{2n}}}$$

ここで x は無次元化した角周波数である。ロウパスフィルタを考えた場合に x は次のようになる。

$$x = c\,\omega T$$

ここでcは定数，ω は角周波数，T はサンプリング周期である。$|x| << 1$ では $|F(x)| \simeq 1$ となり入力とほぼ同じ振幅の出力が得られ，$|x| >> 1$ では $|F(x)| \simeq |x|^{-n}$ となり x が大きくなるにつれて，出力振幅が小さくなる。チェビシェフ型のものは周波数特性がチェビシェフの多項式に従うものである。チェビシェフの多項式は次の通りである。

$$C_n(x) = \begin{cases} \cos(n \cdot \cos^{-1} x) & |x| \leqq 1 \\ \cosh(n \cdot \cosh^{-1} x) & |x| \geqq 1 \end{cases}$$

フィルタの周波数特性は次のとおりである。

$$|F(x)| = \frac{1}{\sqrt{1 + \varepsilon^2 C_n(x)^2}}$$

$|x| \leqq 1$ となる通過域ではF(x)は

$$1/\sqrt{1-\varepsilon^2} \leqq |f(x)| \leqq 1/\sqrt{1+\varepsilon^2}$$

となり振幅が一定範囲内にある。$|x| \geqq 1$ の遮断領域では振幅は急激に減衰する特性となる。

ベッセルフィルタは平坦遅延特性を有するものである。平坦遅延特性とは，フィルタの遅延時間の周波数依存性が小さい特性である。矩形波を入力した場合に，アンダーシュートやオーバーシュートなどを生じにくい特性となる。電気回路では，ディジタル信号処理などに適したものとなっている。

ベッセルフィルタの基礎理論について次に述べる。

* 科学技術庁研究開発局，現所属：気象庁地震火山部地震予知情報課

ベッセルフィルタの伝達関数については，柳沢・神林（1986）に基づいて説明する。一般に伝達関数の逆数T（ix）は次のように表される。

$$T(ix)=g(x)+iu(x)$$

ここでxは無次元化された角周波数である。伝達関数$1/T(ix)$の位相特性$\theta(x)$は次のようになる。

$$\theta(x)=\tan^{-1}\frac{u(x)}{g(x)}$$

ここで目指すフィルタは応答の時間遅れが周波数に依存しないものである。時間遅れが周波数によらないとは，位相遅れが周波数に比例することである。すなわち，時間遅れΔtは

$$\Delta t=\frac{\theta}{x}$$

であるので，$\Delta t=\mathrm{constant}$とするために，位相特性$\theta(x)$を次のようにおく。

$$\theta(x)=\tan^{-1}\frac{u(x)}{g(x)}=x$$

したがって

$$\tan x=\frac{u(x)}{g(x)}$$

次に$\tan x$を連分数に展開する

$$\tan x=\cfrac{1}{\cfrac{1}{x}-\cfrac{1}{\cfrac{3}{x}-\cfrac{1}{\cfrac{5}{x}-\cdots\cdots}}}$$

$$=\cfrac{x}{1+\cfrac{-x^2}{3+\cfrac{-x^2}{5+\cdots\cdots}}}$$

$\tan x$を展開する方式はこれが一意ではないが，ベッセルフィルタの伝達関数を求める上では上記の展開法が用いられている。次にこれを次数を制限して整理する。分母の第1項までとると式は次のようになる。

$$\frac{u_1}{g_1}=x\cdot\frac{1}{1}$$

次に分母の‘1’に1つ次数を増やした式を代入することにより1次高い次数の式を得る。

$$1\leftarrow 1-\frac{x^2}{3}$$

$$\frac{u_2}{g_2}=x\cdot\frac{1}{1-\frac{x^2}{3}}$$

$$=x\cdot\frac{1\cdot 3}{1\cdot 3-x^2}$$

次に‘3’に1つ次数を増やした式を代入する。

$$3\leftarrow 3-\frac{x^2}{5}$$

$$\frac{u_3}{g_3}=x\cdot\frac{1\cdot\left(3-\frac{x^2}{5}\right)}{1\cdot\left(3-\frac{x^2}{5}\right)-x^2}$$

同様にして

$$\frac{u_4}{g_4}=x\cdot\frac{1\cdot3\cdot5\cdot7-(1\cdot3+7)x^2}{1\cdot3\cdot5\cdot7-(1\cdot3+7+5\cdot7)x^2+x^4}$$

$$=x\cdot\frac{105-10x^2}{105-45x^2+x^4}$$

$$\vdots$$

このように漸化的に次数の大きな位相特性が得られる。n－1次の位相特性の式が得られたところで，次の代入を行うことによりn次の位相特性の式が得られる。

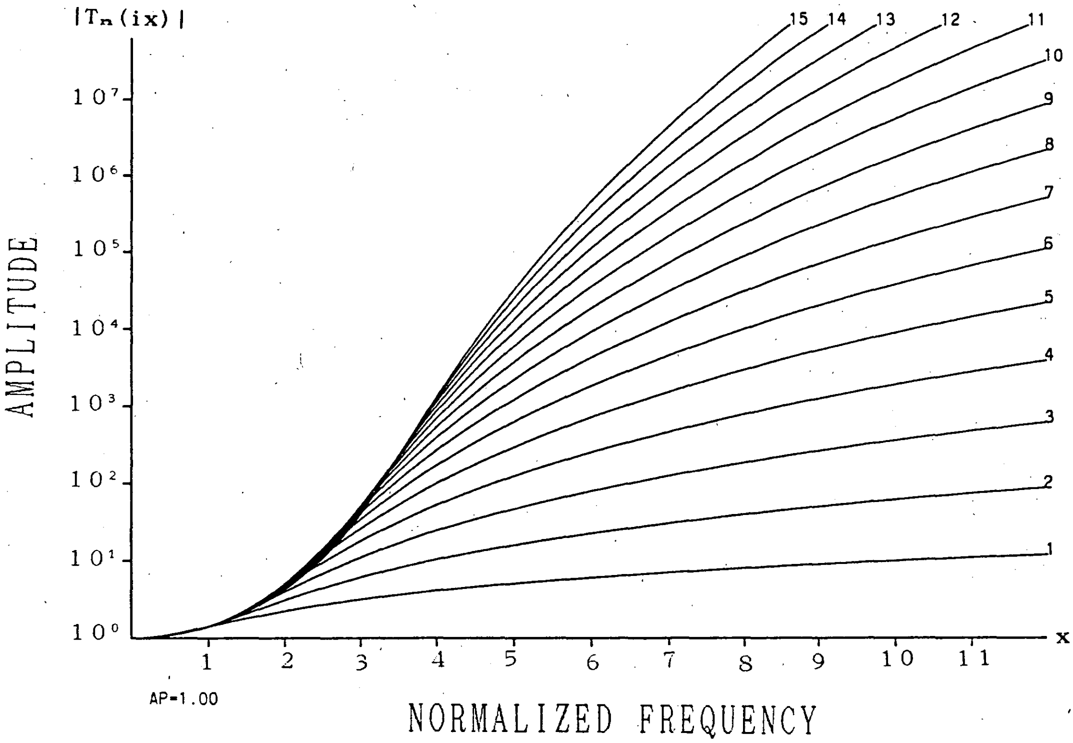
$$(2n-3)\leftarrow(2n-3)-\frac{x^2}{2n-1}$$

この代入操作をプログラム化し，指定した次数のフィルタの位相特性の式が得られるようにした。伝達関数の逆数$T(ix)=g(x)+iu(x)$を，$ix=s$としてラプラス変換の形式になおすと各次数の式は次の様になる。

$$\begin{aligned}
T_1(s)&=s+1\\
T_2(s)&=(s^2+3s+3)/3\\
T_3(s)&=(s^3+6s^2+15s+15)/15\\
T_4(s)&=(s^4+10s^3+45s^2+105s+105)/105\\
T_5(s)&=(s^5+15s^4+105s^3+420s^2+945s\\
&\qquad+945)/945\\
&\vdots
\end{aligned}\tag{1}$$

ここで$x=0$で$|T(ix)|=1$となるように係数をかけている。各次数の振幅特性$|T(ix)|$を第1図に示す。第1図では振幅特性が3 dB変化する（$\sqrt{2}$倍になる）xを1.0として，周波数について規格化している。フィルタの伝達関数は，$T_n(ix)^{-1}$であるのでこれはロウパス特性となっている。

(1)式：$T_n(s)=0$としたときのsの根を第2図に示す。第2図中の数字は式の次数を表す。一つの式から得られた根はわかりやすくするために線で結んである。ここで示した根s_iの全ては$Re(s_i)\leqq 0$となっており，$1/T_n(s)$が物理的に実現可能な特性であることがわかる。



第1図　フィルタの次数と振幅特性
ベッセルフィルタの伝達関数の逆数の絶対値 $\left|T_n(ix)\right|$ を示す．各曲線の数字はフィルタの次数を表す．横軸は $x=1$ において $\left|T_n(ix)\right|=\sqrt{2}$ となるように規格化している．

この伝達関数をディジタルフィルタとして実現するためには，z 変換を施す必要がある。ロウパスフィルタを実現するためには，次の双1次変換が用いられる（Hamming，宮川・今井訳，1980）。

$$x=\frac{c}{i}\cdot\frac{1-z}{1+z}\quad(c>0) \tag{2}$$

ここで，c は無次元化した角周波数と実際の周波数を対応づけるための定数でもある。c は，フィルタのパラメータを与えることにより決まる。フィルタを決定するパラメータとして，カットオフ周波数，カットオフ周波数における減衰量，フィルタの次数を与えることとする。ロウパスフィルタを決めるパラメータとしてはフィルタの次数を与えずに，通過域の最高周波数とその減衰量，遮断域の周波数とその減衰量を与える方が一般的である。しかし，第1図に示すようにベッセルフィルタの場合には次数を上げても高域の減衰量が急激には大きくならないので，数値計算における不都合が生ずることを避けるため，フィルタ次数をパラメータとして与えることとした。高域の減衰量については，第1図からも求められる。

通過域における振幅の最小を斎藤（1978）にならい，$1/(1+A_p^2)^{1/2}$とする。減衰量に関するパラメータ A_pを与えると

$$\frac{1}{(1+A_p^2)^{1/2}}=\left|\frac{1}{T_n(ix_p)}\right| \tag{3}$$

となる x_pが得られる（第3図）。z については，定義より次の通りである。

$$z=\exp(-i\omega T) \tag{4}$$

ここで ω は角周波数，Tはサンプリング周期である。通過域の角周波数の最高を ω_pとし，それを(2)式に代入する。

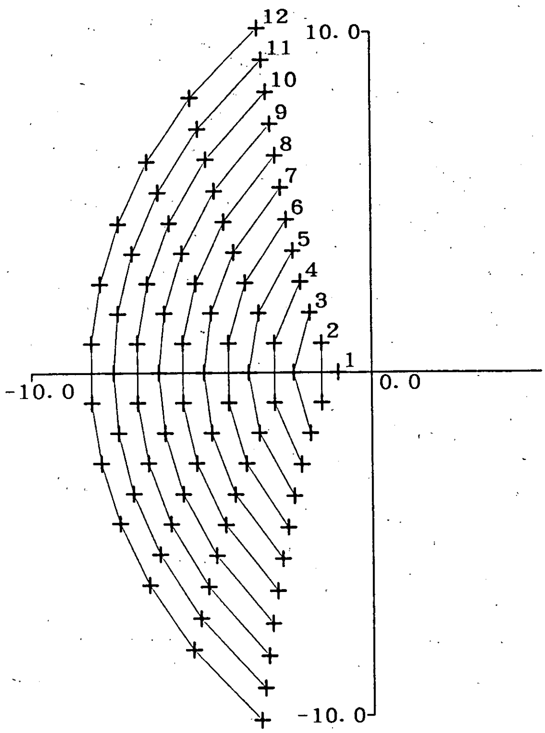
$$x_p=\frac{c}{i}\cdot\frac{1-\exp(-i\omega_p T)}{1+\exp(-i\omega_p T)}$$
$$=\mathrm{ctan}(\omega_p T/2)$$

この式より周波数比を決める定数 c が得られる。

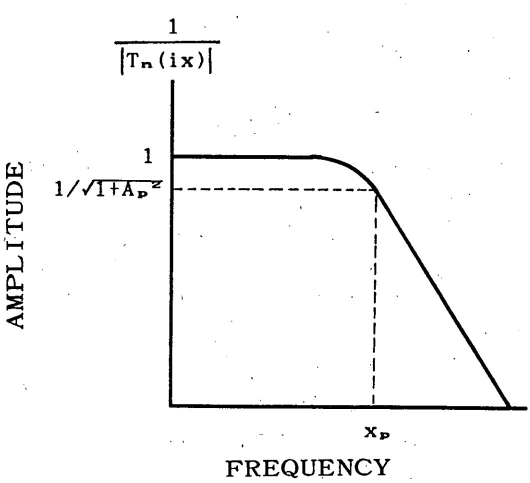
$$c=x_p\cot(\omega_p T/2) \tag{5}$$

(2)式，(5)式を(1)式に代入することによりディジタルフィルタの係数が決まる。

しかし，上記の $T_n(s)$ にそのまま代入すると式の次数が高くなるにつれて計算が煩雑となる。そこで，斉



第2図　ベッセルフィルタの伝達関数の根
(1)式：$T_n(s)=0$とした時のsの根を示す．グラフの横軸は実部，縦軸は虚部である．数字は式の次数を表す．一つの式から得られた根は線で結んでいる．



第3図　通過域における最小の振幅に関するパラメータ：A_p
フィルタを設計する上で，通過域の最小の振幅に関するパラメータA_pを与える．通過域の最小の振幅は$1/\sqrt{1+A_p{}^2}$となる．

藤（1978）に従い，必要とする特設は次のような分子・分母ともzの2次式となる特性の異なる複雑の基本的なフィルタの積として得る。

$$F(z)=\frac{1+a_{1i}z+a_{2i}z^2}{1+b_{1i}z+b_{2i}z^2}$$

この基本フィルタ$f_i(z)$の数値計算の処理は次のようになる。

$$y_t=x_t+a_{1i}x_{t-1}+a_{2i}x_{t-2}-b_{1i}y_{t-1}-b_{2i}y_{t-2} \quad (6)$$

ここで，x_tは入力，y_tは出力，$t-1$はtより1サンプル間隔前のデータを表す。フィルタ全体としては次のようになる。

$$F(z)=G_n\prod_i F_i(z)$$

最初の基本フィルタの出力値を次のフィルタの入力値として，次々に基本フィルタを直列的に処理することにより，最終的出力が得られる。このようにすることにより数値計算上も安定する。複数の基本フィルタを繰り返して処理するためには，斎藤（1978）によるサブルーチンTANDEM，RECFILの使用が便利である。

伝達関数を(7)式のZ変換の形式にするためには，(1)式の$T_n(s)$を因数分解する必要がある。因数分解は，

$$T_n(s)=0$$

となる高次代数方程式を解き，sの根si（$i=1$，2‥，n）を求めることと同等である。ベアストウーヒチコック（Bairstow−Hitchcock）法（戸川，1981）を用いてこの代数方程式の根を求めることとした。因数分解により伝達関数は次のようになる。

$$\frac{1}{T_n(s)}=\frac{s_1}{(s-s_1)}\cdot\frac{s_1{}^*}{(s-s_1{}^*)}\cdot\frac{s_2}{(s-s_2)}\cdot\frac{s_2{}^*}{(s-s_2{}^*)}\cdots\cdots$$

周波数0（$s=0$）で振幅が1となるようにするために，根を分子にかけている。第2図に示すように実根は，式が奇数次のときに1つあるだけで，他は共役な複素根の対をなしている。共役の部分を取り出し，そこに(2)式を代入する。

$$\frac{s_i\cdot s_i{}^*}{(s-s_i)(s-s_i{}^*)}=\frac{|s_i|^2}{c^2\dfrac{(1-z)^2}{(1+z)^2}-2Re(s_i)c\dfrac{1-z}{1+z}+|s_i|^2}$$

$$=G_i\frac{1+a_{1i}z+a_{2i}z^2}{1+b_{1i}z+b_{2i}z^2}$$

ここで

$$G_i=\frac{|s_i|^2}{a}$$

$$\alpha=c^2-2Re(s_i)\ c+|s_i|^2$$

$a_{1i} = 2$

$a_{2i} = 1$

$b_{1i} = (-2c^2 + 2|s_i|^2)/\alpha$

$b_{2i} = (c^2 + 2Re(s_i)c + |s_i|^2)/\alpha$

フィルタの次数が奇数の場合，sは実数の根を1つ含むがそれについては次のようになる。

$$G_i = \frac{-s_i}{c - s_i}$$

$a_{1i} = 1$

$a_{2i} = 0$

$$b_{1i} = \frac{-c - s_i}{c - s_i}$$

$b_{2i} = 0$

iについてフィルタ処理を繰り返して施すことにより，必要とする特性を得る。ここで，以上の処理手順をまとめておく。

①通過域の角周波数の最高ω_p，通過域における振幅の最小$1/(1 + A_p{}^2)^{1/2}$，フィルタの次数nを与える。

②(3)式によりA_pからx_pを求める。

③(5)式によりω_p，x_pからcを求める。

④(1)式を因数分解する。

⑤因数分解した各因数式に(2)，(5)式を代入して，フィルタの係数G_i，a_{1i}，a_{2i}，b_{1i}，b_{2i}を求める。

このようにして得られるディジタルフィルタの周波数特性はz変換の定義に従い，$z = \exp(-i\omega T)$を式に代入することにより求められる。第4図にサンプリング周期：$T = 0.01$〔sec〕，通過域の最高周波数：$f_p = 1.0$〔Hz〕($\omega_p = 2\pi f_p$)，フィルタの次数：$n = 4$，減衰量に関するパラメータ：$A_p = 1.0$（3 dBの減衰）を与えた場合の振幅及び位相の周波数特性の例を示す。ここで位相は位相遅れを正にとってある（第4，6，7，8も同様）。このフィルタによる矩形波に対する時間応答を第5図に示す。比較のためにバタワースフィルタによる時間応答も併せて示す。ここで使用したバタワースフィルタの周波数特性は第6図に示すもので，フィルタの次数は4次である。第5図に示すようにベッセルフィルタは時間遅れの周波数依存性が小さいために，余分な「相」が現れにくい。

次にハイパスフィルタについて記す。ハイパスフィルタを得るためには，(1)式に次のz変換を施す。

$$x = \frac{c}{i} \cdot \frac{1 + z}{1 - z} \quad (c > 0)$$

このz変換によると，$\omega_p \geqq 0$の領域は$x_p \leqq 0$へ，$\omega_p \leqq 0$の領域は$x_p 0 \geqq$へ投影される。ロウパスフィルタの場合と同様に，通過域の最低角周波数をω_p，そこにおける減衰量に関するパラメータをA_p，フィルタの次数を与えるとcは次のようになる。

$$c = -x_p \tan(\omega_p T/2)$$

x_pはロウパスフィルタと同様に(3)式により与えられる。フィルタの各定数は同様に次のようになる。

$$G_i = \frac{|s_i|^2}{\alpha}$$

$\alpha = c^2 - 2Re(s_i)c + |s_i|^2$

$a_{1i} = -2$

$a_{2i} = 1$

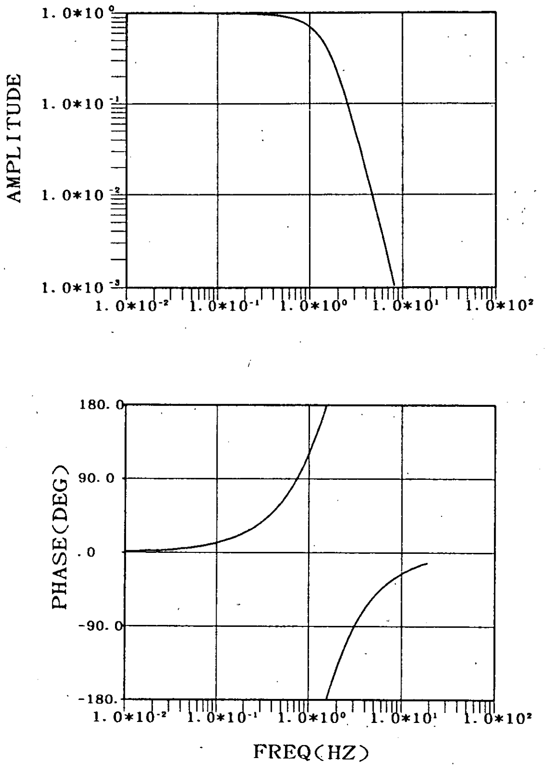
$b_{1i} = (+2c^2 - 2|s_i|^2)/\alpha$

$b_{2i} = (c^2 + 2Re(s_i)c + |s_i|^2)/\alpha$

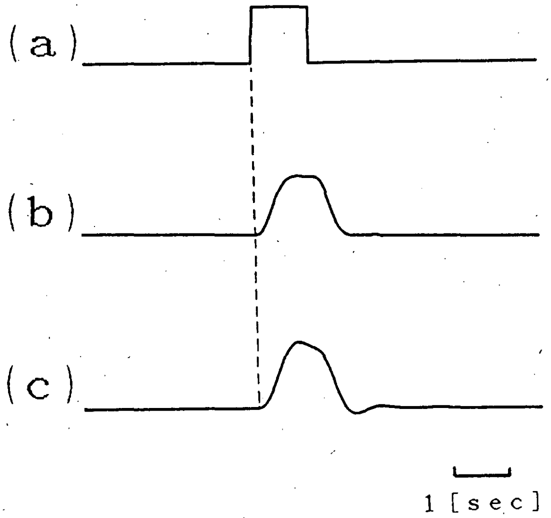
奇数次フィルタの実数根については，次のようになる。

$$G_i = \frac{-s_i}{c - s_i}$$

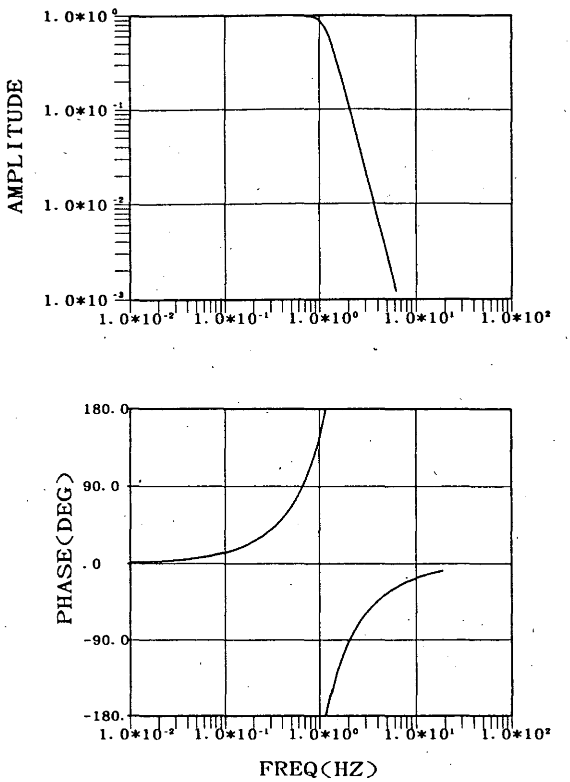
$a_{1i} = -1$



第4図　設計フィルタの周波数特性の例（ロウパスフィルタ）
次数：4，通過域の最高周波数：1.0〔Hz〕
通過域の最大減衰量の係数（A_p）：1.0，サンプル間隔：0.01〔sec〕
位相は遅れを正としている．



第5図　矩形波に対する応答
(a)入力
(b)ベッセルフィルタによる出力
周波数特性は第4図のものである。ベッセルフィルタはオーバーシュートやアンダーシュートを生じにくい.
(c)バタワースフィルタによる出力
周波数特性は第6図のものである.



第6図　バタワースフィルタの周波数特性
第5図(c)に示す時間応答に用いたロウパス・バタワース・フィルタの特性
次数：4，通過域の最高周波数：1.0〔Hz〕，通過域の最大減衰量の係数（Ap）：≦1.0
サンプル間隔：0.01〔sec〕
位相は遅れを正としている.

$$a_{2i}=0$$

$$b_{1i}=\frac{c+s_i}{c-s_i}$$

$$b_{2i}=0$$

第7図にサンプリング周期：$T=0.01$〔sec〕，通過域の最低周波数：$f_p=1.0$〔Hz〕（$\omega_p=2\pi f_p$），フィルタの次数：$n=4$，減衰量に関するパラメータ：$A_p=1.0$を与えた場合の振幅の周波数特性の例を示す。

次にバンドパスフィルタについて記す。バンドパスフィルタの場合には，次の式によりxをまず，λへ変換する。

$$x=\frac{\lambda^2-\lambda_o^2}{\lambda} \tag{8}$$

この変換により，$x\leqq x_p$となる通過域は$\lambda_L\leqq\lambda\leqq\lambda_H$（$\lambda_L$，$\lambda_H>0$）へ変換される。これはバンドパスの特性となっている。なお，λ_L，λ_Hは$x=\pm x_p$としたときのλの根である。この(8)式よりx_p，λ_L，λ_Hの間には次の関係があることがわかる。

$$x_p=\lambda_H-\lambda_L$$
$$\lambda_o^2=\lambda_L\lambda_H \tag{9}$$

λとzとの変換はロウパスフィルタと同様の双一次変換による。

$$\lambda=\frac{c}{i}\cdot\frac{1-z}{1+z}$$

通過域の高域側の角周波数をω_H，低域側の角周波数をω_Lとすると，λ_L，λ_Hは(4)式を代入することにより次のようになる。

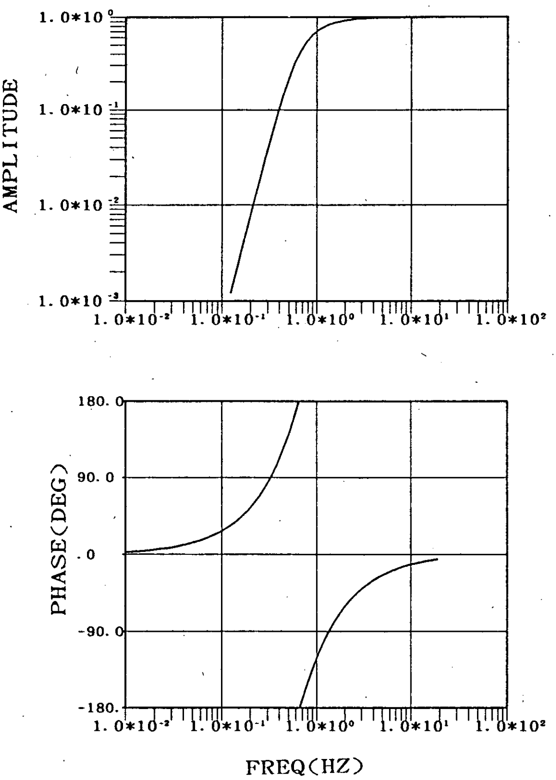
$$\lambda_H=c\cdot\tan(\omega_H T/2),$$
$$\lambda_L=c\cdot\tan(\omega_L T/2) \tag{10}$$

(9)，(10)式より，特性関数の通過域の変数x_pと通過帯域の角周波数ω_H，ω_Lの関係は

$$x_p=\lambda_H-\lambda_L$$
$$=c\cdot(\tan(\omega_H T/2)-\tan(\omega_L T/2))$$

となる。上式より周波数比に関する定数cが求められる。またλ_oについては

$$\lambda_o^2=\lambda_L\lambda_H$$
$$=c^2\tan(\omega_H T/2)\tan(\omega_L T/2)$$



第７図　設計フィルタの周波数特性の例（ハイパスフィルタ）
次数：4，通過域の最低周波数1.0〔Hz〕，通過域の最大減衰量の係数（A_p）：1.0
サンプル間隔：0.01〔sec〕
位相は遅れを正としている．

(8)式を伝達関数の式に代入し，整理する。

$$\frac{s_i\cdot s_i^*}{(s-s_i)(s-s_i^*)}$$

$$=\frac{|s_i|^2(i\lambda)^2}{\{(i\lambda)^2-(i\lambda)s_i+\lambda_o^2\}\{(i\lambda)^2-(i\lambda)s_i^*+\lambda_o^2\}}$$

$$=\frac{|s_i|^2(i\lambda)^2}{(i\lambda-\lambda_{i1})(i\lambda-\lambda_{i2})(i\lambda-\lambda_{i1}^*)(i\lambda-\lambda_{i2}^*)}$$

$$=\frac{|s_i|(i\lambda)}{(i\lambda-\lambda_{i1})(i\lambda-\lambda_{i1}^*)}\cdot\frac{|s_i|(i\lambda)}{(i\lambda-\lambda_{i2})(i\lambda-\lambda_{i2}^*)}\quad(11)$$

ここで

$$\lambda_{i1}=\frac{s_i+\sqrt{s_i^2-4\lambda_o^2}}{2},\quad\lambda_{i2}=\frac{s_i-\sqrt{s_i^2-4\lambda_o^2}}{2}$$

このような式の変形を行わず，(11)式をそのまま展開して z についての４次式を求めてもよいが，そのときには数値計算上の精度不足を発生する場合がある。(11)式は以下のように変形される。

$$(11)式=\frac{1-z^2}{a_{01i}+a_{11i}z+a_{21i}z^2}\cdot\frac{1-z^2}{a_{02i}+a_{12i}z+a_{22i}z^2}$$

ここで

$$a_{01i}=\frac{c}{|s_i|}-\frac{2Re(\lambda_{i1})}{|s_i|}+\frac{|\lambda_{i1}|^2}{c|s_i|}$$

$$a_{11i}=\frac{-2c}{|s_i|}+\frac{2|\lambda_{i1}|^2}{c|s_i|}$$

$$a_{21i}=\frac{c}{|s_i|}+\frac{2Re(\lambda_{i1})}{|s_i|}+\frac{|\lambda_{i1}|^2}{c|s_i|}$$

$$a_{02i}=\frac{c}{|s_i|}-\frac{2Re(\lambda_{i2})}{|s_i|}+\frac{|\lambda_{i2}|^2}{c|s_i|}$$

$$a_{12i}=\frac{-2c}{|s_i|}+\frac{2|\lambda_{i2}|^2}{c|s_i|}$$

$$a_{22i}=\frac{c}{|s_i|}+\frac{2Re(\lambda_{i2})}{|s_i|}+\frac{|\lambda_{i2}|^2}{c|s_i|}$$

なお，奇数次フィルタの実数根については，次のようになる。

$$\frac{-s_i}{s-s_1}=\frac{1-z^2}{a_{0i}+a_{1i}z+a_{2i}z}$$

ここで

$$a_{0i}=\frac{c}{s_i}-\frac{\lambda_o^2}{cs_i}+1$$

$$a_{1i}=\frac{2c}{s_i}-\frac{2\lambda_o^2}{cs_i}$$

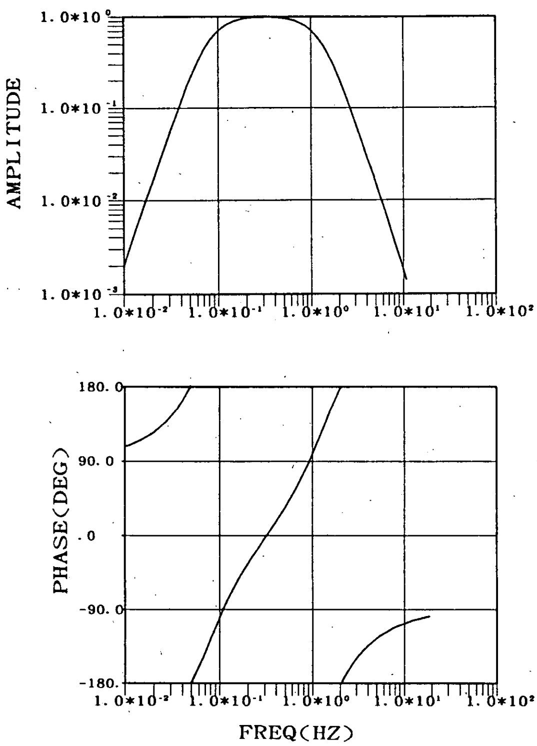
$$a_{2i}=\frac{c}{s_i}-\frac{\lambda_o^2}{cs_i}-1$$

第８図にサンプリング周期：$T=0.01$〔sec〕，通過域の最高周波数：$f_H=1.0$〔Hz〕（$\omega_H=2\pi f_H$），通過域の最低周波数：$f_L=0.1$〔Hz〕（$\omega_L=2\pi f_L$），フィルタの次数：$n=3$，減衰量に関するパラメータ：$A_p=1.0$ を与えた場合の振幅および位相の周波数特性の例を示す。

以上の３タイプのディジタルフィルタのプログラムリスト（FORTRAN）をAppendixに示す。基本フィルタを繰り返して処理するための斉藤（1978）によるサブルーチンTANDEM，RECFILも併せてAppendixに掲載する。フィルタの通過帯域の分類として上記の３種の他にバンドエリミネーションフィルタがあるが，地震データ処理における使用頻度は低いと考え，プログラム化は行っていない。なお，バンドエリミネーションフィルタは，バンドパスフィルタにおける(8)式の代わりに次の式により周波数変換を行うことにより，実現できる。

$$x=\frac{\lambda}{\lambda^2-\lambda_o^2}$$

ロウパスフィウタのプログラムにおけるルーチン構造を第９図に示しておく。ここで各サブルーチンの機能は



第８図　設計フィルタの周波数特性の例（バンドパスフィルタ）
次数：4，通過域の周波数：0.1～1.0〔Hz〕，
通過域の最大減衰量の係数（A_p）：1.0
サンプル間隔：0.01〔sec〕
位相は遅れを正としている.

次の通りである。

BESLPN ：与えられた条件の下におけるベッセルフィルタの係数を得る。

GETBPL ：与えられた次数のベッセルフィルタの伝達関数の基本式（(1)式）を得る。

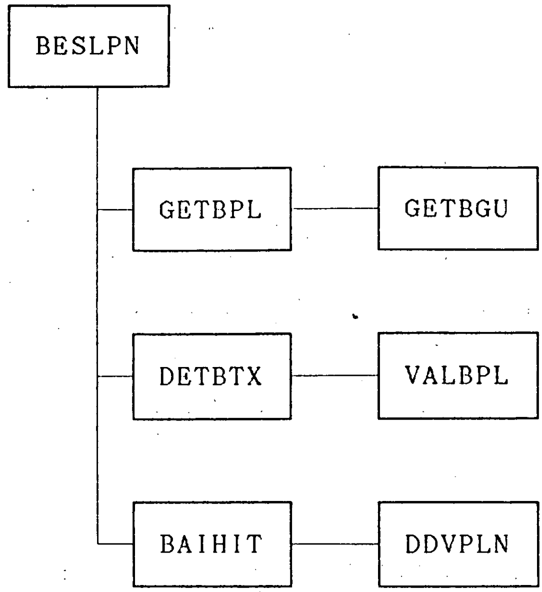
GETBGU ：与えられた次数の伝達関数の基本式（(1)式）の実数部と虚数部をそれぞれ独立に得る。

DETBTX ：伝達関数が与えられた減衰値になるときのxを求める。

VALBPL ：与えられた入力値における伝達関数の振幅を得る。

BAIHIT ：ベアストウーヒッチコック法により高次代数方程式の根を得る。

DDVPLN ：高次式を２次式により割る。



第９図　ロウパスフィルタのプログラムのルーチン構造

ハイパス，バンドパスフィルタについてもルーチン構造は同等である。

§3.　おわりに

位相特性がよいとされているベッセルタイプのディジタルフィルタの自動設計方式を示した。ただ単に位相遅れをなくするためには，斉藤（1978）が述べているように，ディジタルフィルタを正方向と逆方向に処理するか，ＦＩＲ型のフィルタを用いれば達成される。ここで示したフィルタは，遅延特性を重視する場合であり，かつ因果律を満たした物理的に実現可能な特性が必要であるかリアルタイム的処理を行う場合に有効であると考える。

参考文献

斎藤　正徳（1978）：漸化式ディジタル・フィルターの自動設計，物理探査，**31**，240－263.

戸川　隼人（1981）：数値計算法，コロナ社，95－97.

柳沢　　健・神林　紀嘉（1986）：フィルタの理論と設計，秋葉出版株式会社，132－137.

Hamming，R. W.（1980）：ディジタル・フィルタ，宮川　洋・今井　秀樹訳，科学技術出版社，230 pp.

APPENDIX　プログラムリスト

1 . BESLPN

ベッセル・ロウパス・ディジタルフィルタの係数を求める．

引数　H(I)：フィルタの係数（出力）

M：フィルタの次数（出力）

GN：フィルタの倍率係数（出力）

フィルタ特性は次の式による

$$F(z)=G_N\prod_{i=1}^{M}\frac{1+H_{(4i-3)}z+H_{(4i-2)}z^2}{1+H_{(4i-1)}z+H_{(4i)}z^2}$$

（実際の計算処理は，本文(6)式参照）

N：ベッセルフィルタの次数（入力）

FPK：通過域の最高周波数［Ｈｚ］（入力）

AP：通過域の最小振幅に関するパラメータ（入力）

通過域の最小振幅は

$$\frac{1}{(1+A_P{}^2)^{1/2}}$$

となる．

DT：サンプリング間隔［SEC］（入力）

IERR：≠０のときエラー（出力）

```
C==================================================================
      SUBROUTINE BESLPN (H,M,GN,N,FPK,AP,DT,IERR)
C -----------------------------------------------------------------
C       BESSEL LOW PASS FILTER COEFFICIENT
C -----------------------------------------------------------------
C **** ARGUMENTS*****
C ( O )  H        : FILTER COEFFICIENTS
C ( O )  M        : ORDER OF FILTER   (M=(N+1)/2)
C ( O )  GN       : GAIN FACTOR
C ( I )  N        : ORDER OF BESSEL FILTER FUNCTION
C ( I )  FPK      : PASS BAND FREQUENCY  (DIMENSIONAL)   ( HZ )
C ( I )  AP       : MAX. ATTENUATION IN PASS BAND
C ( I )  DT       : SAMPLING INTERVAL
C -----------------------------------------------------------------
C        A. KATSUMATA ( 1989/5/20 )
C -----------------------------------------------------------------
      DIMENSION H(*)
      DIMENSION COEFBS(0:50),WCOEF(0:50)
      COMPLEX   ZROOT(50)
      DATA  PI/3.141593/,HP/1.570796/
C **** ARGUMENTS  *****
      FP = FPK * DT
      WP = ABS(FP)*PI
      IF( WP.EQ.0. .OR. WP.GE.HP )  GO TO  100
C****  DETERMINE  C  ******
      TP = TAN(WP)
      PA = ABS(AP)
      IF( PA.EQ.0. )  PA = 0.5
C----- GET BESSEL FILTER POLYNOMIAL
      CALL GETBPL(N,COEFBS(0),MAXDBS,IERROR)
      IF(IERROR.NE.0) GOTO 100
      CALL DETBTX(PA,COEFBS(0),MAXDBS,TX,IERROR)
      IF(IERROR.NE.0) GOTO 100
      C=TX/TP
      W1=1.0
      IF(COEFBS(MAXDBS).LT.0.99999.OR.COEFBS(MAXDBS).GT.1.00001)
     +              W1=1.0/COEFBS(MAXDBS)
C
      DO 20 J=0,MAXDBS-1
         WCOEF(MAXDBS-J)=COEFBS(J)*W1
   20 CONTINUE
C------ GET ROOTS OF POLYNAMIAL
      CALL BAIHIT(MAXDBS,WCOEF(1),ZROOT(1))
C -----------------------------------------------------------------
      M   = MAXDBS/2
      G   = 1.
C -----------------------------------------------------------------
      DO 1 J=1,M
        SR = REAL(ZROOT(2*J-1))
        SI = AIMAG(ZROOT(2*J-1))
        SA2= SR*SR+SI*SI
        A   = 1./(SA2+C*C-C*2.0*SR)
        G   = G*A*SA2
          H(J*4-3) = 2.
          H(J*4-2) = 1.
          H(J*4-1) = (2.0*SA2-2.0*C*C)*A
          H( J*4 ) = (C*C+SA2+C*2.0*SR)*A
    1 CONTINUE
C****  FOR ODD N  *****
      IF( MOD(MAXDBS,2).NE.0 ) THEN
        M = M + 1
        SR=REAL(ZROOT(MAXDBS))
        A = 1.0/(C-SR)
        G = G*A*(-SR)
         H(M*4-3) = 1.
         H(M*4-2) = 0.
         H(M*4-1) = (-C-SR)*A
         H( M*4 ) = 0.
      ENDIF
      GN = G
      IERR=0
      RETURN
C****  ERROR  ******
  100 WRITE(6,101)  FP
  101 FORMAT(/1X,5('?'),'   (BESLPN)   INVALID INPUT   FP =',
     *        1PE14.6,3X,5('?')//)
      IERR=1
      RETURN
      END
```

2 . BESHPN

ベッセル・ハイパス・ディジタルフィルタの係数を求める．

引数　H(I)：フィルタの係数（出力）

M：フィルタの次数（出力）

GN：フィルタの倍率係数（出力）

フィルタ処理は次の式による

$$F(z)=G_N \prod_{i=1}^{M} \frac{1+H_{(4i-3)}z+H_{(4i-2)}z^2}{1+H_{(4i-1)}z+H_{(4i)}z^2}$$

N：ベッセルフィルタの次数（入力）
FPK：通過域の最低周波数［Hz］（入力）
AP：通過域の最小振幅に関するパラメータ（入力）
　通過域の最小振幅は

$$\frac{1}{(1+A_P{}^2)^{1/2}}$$

　となる．
DT：サンプリング間隔［SEC］（入力）
IERR：≠０のときエラー（出力）

```
C=================================================================
      SUBROUTINE BESHPN (H,M,GN,N,FPK,AP,DT,IERR)
C ----------------------------------------------------------------------
C      BESSEL HIGH PASS FILTER COEFFICIENT
C ----------------------------------------------------------------------
C **** ARGUMENTS*****
C ( O )  H      : FILTER COEFFICIENTS
C ( O )  M      : ORDER OF FILTER  (M=(N+1)/2)
C ( O )  GN     : GAIN FACTOR
C ( I )  N      : ORDER OF BESSLE FILTER FUNCTION
C ( I )  FPK    : PASS BAND FREQUENCY  (DIMENSIONAL)  ( HZ )
C ( I )  AP     : MAX. ATTENUATION IN PASS BAND
C ( I )  DT     : SAMPLING INTERVAL (SEC)
C ----------------------------------------------------------------------
C      A. KATSUMATA (1989/5/20)
C ----------------------------------------------------------------------
      DIMENSION H(*)
      DIMENSION COEFBS(0:50),WCOEF(0:50)
      COMPLEX   ZROOT(50)
      DATA  PI/3.141593/,HP/1.570796/
C
C------ NON-DIMENSIONAL FREQUENCY
      FP = FPK       * DT
      WP = ABS(FP)*PI
      IF( WP.EQ.0..OR. WP.GE.HP)  GO TO  100
C****  DETERMINE C  *****
      TP = TAN(WP)
      PA = ABS(AP)
      IF( PA.EQ.0. )  PA = 0.5
C----- GET BESSEL FILTER POLYNOMIAL
      CALL GETBPL(N,COEFBS(0),MAXDBS,IERROR)
      IF(IERROR.NE.0) GOTO 100
      CALL DETBTX(PA,COEFBS(0),MAXDBS,TX,IERROR)
      IF(IERROR.NE.0) GOTO 100
      C  = TX*TP
C----- GET BESSEL FILTER POLYNOMIAL ---------------------
      CALL GETBPL(N,COEFBS(0),MAXDBS,IERROR)
      W1=1.0
      IF(COEFBS(MAXDBS).LT.0.99999.OR.COEFBS(MAXDBS).GT.1.00001)
     +              W1=1.0/COEFBS(MAXDBS)
C
      DO 20 J=0,MAXDBS-1
        WCOEF(MAXDBS-J)=COEFBS(J)*W1
   20 CONTINUE
C------ GET ROOTS OF POLYNAMIAL
      CALL BAIHIT(MAXDBS,WCOEF(1),ZROOT(1))
C ----------------------------------------------------------------------
      M  = MAXDBS/2
      G  = 1.
C ----------------------------------------------------------------------
      DO 1 J=1,M
        SR = REAL(ZROOT(2*J-1))
        SI = AIMAG(ZROOT(2*J-1))
        SA2= SR*SR+SI*SI
        A  = 1./(SA2+C*C-C*2.0*SR)
        G  = G*A*SA2
          H(J*4-3) = -2.
          H(J*4-2) = 1.
          H(J*4-1) =-(2.0*SA2-2.0*C*C)*A
          H( J*4 ) = (C*C+SA2+C*2.0*SR)*A
    1 CONTINUE
      IF( MOD(MAXDBS,2).NE.0 ) THEN
C****  FOR ODD N  *****
        M = M + 1
        SR=REAL(ZROOT(MAXDBS))
        A = 1.0/(C-SR)
        G = G*A*(-SR)
         H(M*4-3) = -1.
         H(M*4-2) = 0.
         H(M*4-1) =-(-C-SR)*A
         H( M*4 ) = 0.
      ENDIF
      GN = G
      IERR=0
      RETURN
C****  ERROR  *****
  100 WRITE(6,101)  FP
  101  FORMAT(/1X,5('?'),'   (BESHPN)   INVALID INPUT   FP =',
     *          1PE14.6,3X,5('?')//)
      IERR=1
      RETURN
      END
```

３．BESPSN
　ベッセル・バンドパス・ディジタルフィルタの係数を求める．
　引数　H(I)：フィルタの係数（出力）
　　　　M：フィルタの次数（出力）
　　　　GN：フィルタの倍率係数（出力）
　　　　　フィルタ処理は次の式による

$$F(z)=G_N \prod_{i=1}^{M} \frac{1+H_{(4i-3)}z+H_{(4i-2)}z^2}{1+H_{(4i-1)}z+H_{(4i)}z^2}$$

　　　　N：ベッセイルフィルタの次数（入力）
　　　　FLK：通過域の最低周波数［Hz］（入力）
　　　　FHK：通過域の最高周波数［Hz］（入力）
　　　　AP：通過域の最小振幅に関するパラメータ（入力）

通過域の最小振幅は

$$\frac{1}{(1+A_P{}^2)^{1/2}}$$

となる．

DT：サンプリング間隔［SEC］（入力）

IERR：≠０のときエラー（出力）

```
C=====================================================================
      SUBROUTINE BESPSN (H,M,GN,N,FLK,FHK,AP,DT,IERR)
C ---------------------------------------------------------------------
C       BESSEL BAND PASS FILTER COEFFICIENT
C ---------------------------------------------------------------------
C **** ARGUMENTS*****
C ( O )  H        : FILTER COEFFICIENTS
C ( O )  M        : ORDER OF FILTER   (M=(N+1)/2)
C ( O )  GN       : GAIN FACTOR
C ( I )  N        : ORDER OF BESSEL FILTER FUNCTION
C ( I )  FLK      : LOW  FREQUENCY CUT-OFF  (DIMENSIONAL) ( HZ )
C ( I )  FHK      : HIGH FREQUENCY CUT-OFF  (DIMENSIONAL) ( HZ )
C ( I )  AP       : MAX. ATTENUATION IN PASS BAND
C ( I )  DT       : SAMPLING INTERVAL
C ---------------------------------------------------------------------
C       A. KATSUMATA (1989/5/22)
C ---------------------------------------------------------------------
      DIMENSION H(*)
      DIMENSION COEFBS(0:50),WCOEF(0:50)
      COMPLEX  ZROOT(50)
     +         ,SIO,SIC,RO2,CR1,CR2
C
      DATA  PI/3.141593/,HP/1.570796/
      FL = FLK        * DT
      FH = FHK        * DT
      WL = AMIN1(ABS(FL),ABS(FH))*PI
      WH = AMAX1(ABS(FL),ABS(FH))*PI
      PA = ABS( AP )
      IF( PA.EQ.0. )  PA = 0.5
C
      IF( WL.LE.0. .OR. WL.EQ.WH .OR. WH.GE.HP ) GOTO 100
C
C----- GET BESSEL FILTER POLYNOMIAL ----------------------
      CALL GETBPL(N,COEFBS(0),MAXDBS,IERROR)
      IF(IERROR.NE.0) GOTO 100
C
      CALL DETBTX(PA,COEFBS(0),MAXDBS,TX,IERROR)
      IF(IERROR.NE.0) GOTO 100
C****  DETERMINE  C *****
      C=TX/(TAN(WH)-TAN(WL))
      WA = TAN(WL)*TAN(WH)
C----------------------------------------------------------------
      W1=1.0
      IF(COEFBS(MAXDBS).LT.0.99999.OR.COEFBS(MAXDBS).GT.1.00001)
     +           W1=1.0/COEFBS(MAXDBS)
      DO 20 J=0,MAXDBS-1
         WCOEF(MAXDBS-J)=COEFBS(J)*W1
   20 CONTINUE
C------ GET ROOTS OF POLYNAMIAL
      CALL BAIHIT(MAXDBS,WCOEF(1),ZROOT(1))
C---------------------------------------------------------------
      K  = MAXDBS/2
      M  = K*2
      L  = 0
      G  = 1.
C ----------------------------------------------------------------
      DO  2  J=1,K
         SIO = ZROOT(J*2-1)
         SIC = ZROOT(J*2)
         ZABS = SQRT(SIO*SIC)
         RO2 = CMPLX(WA,0.0)*CMPLX(C,0.0)*CMPLX(C,0.0)
         CR1 = (SIO + CSQRT( SIO*SIO -4.0*RO2)) / 2.0
         CR2 = (SIO - CSQRT( SIO*SIO -4.0*RO2)) / 2.0
C
         A21 = C/ZABS+2.0*REAL(CR1)/ZABS+CABS(CR1)*CABS(CR1)/C/ZABS
         A11 = -2.0*C/ZABS+2.0*CABS(CR1)*CABS(CR1)/C/ZABS
         A01 = C/ZABS-2.0*REAL(CR1)/ZABS+CABS(CR1)*CABS(CR1)/C/ZABS
C
         A22 = C/ZABS+2.0*REAL(CR2)/ZABS+CABS(CR2)*CABS(CR2)/C/ZABS
         A12 = -2.0*C/ZABS+2.0*CABS(CR2)*CABS(CR2)/C/ZABS
         A02 = C/ZABS-2.0*REAL(CR2)/ZABS+CABS(CR2)*CABS(CR2)/C/ZABS
C
         H(J*8-7) = 0.0
         H(J*8-6) =-1.0
         H(J*8-5) = A11/A01
         H(J*8-4) = A21/A01
         H(J*8-3) = 0.0
         H(J*8-2) =-1.0
         H(J*8-1) = A12/A02
         H(J*8  ) = A22/A02
         G = G/(A01*A02)
    2 CONTINUE
C****  FOR ODD N  ******
      IF( MAXDBS.NE.M ) THEN
         M = M + 1
         WB=C/REAL(ZROOT(MAXDBS))
         WW0=-WA*WB-WB+1.0
         WW1=-2.0*(WA-1.0)*WB
         WW2=-WA*WB-WB-1.0
         G  = G / WW0
          H(M*4-3) = 0.
          H(M*4-2) =-1.
          H(M*4-1) = WW1/WW0
          H(M*4  ) = WW2/WW0
      ENDIF
      GN = G
      IERR=0
      RETURN
C****  ERROR  *****
  100 WRITE(6,101)  FL,FH
  101 FORMAT(/1X,5('?'),'   (BESPSN)   INVALID INPUT   FL =',
     *        1PE14.6,3X,'FH =',E14.6,3X,5('?')//)
      IERR=1
```

```
      RETURN
      END
```

4 . GETBPL
フィルタの次数を与えて伝達関数のもとになる多項式を得る.
引数　NN : 次数（入力）
COEFBS(N) : 多項式の係数, Nは次数（≧0）（出力）
MAXBS : 多項式における最大次数（出力）
IERR : ≠0のときエラー（出力）

```
C**********************************************
C
      SUBROUTINE GETBPL(NN,COEFBS,MAXBS,IERR)
C
C      GET THE BESSEL FILTER POLYNOMIAL
C
C       NN     :  ORDER OF POLYNOMIAL
C       COEFBS :  COEFFICIENT OF POLYNOMIAL
C       MAXBS  :  MAXIMUM ORDER OF POLYNOMIAL
C       IERR   :  .NE.0 ==> ERROR
C
C**********************************************
C
      DIMENSION COEFBS(0:50)
C
C       COEFG  : COEFFICIENT OF POLYNOMIAL  ( FOR REAL PART )
C       COEFU  : COEFFICIENT OF POLYNOMIAL  ( FOR IMAGINARY PART )
      DIMENSION COEFG(0:50),COEFU(0:50)
C
C==== CLEAR OUTPUT BUFFER =======================
      DO 10 I=0,50
         COEFBS(I)=0.0
   10 CONTINUE
      CALL GETBGU(NN,COEFG(0),MAXDG,COEFU(0),MAXDU,IERROR)
      IF(IERROR.NE.0) THEN
         IERR=IERROR
         RETURN
      ENDIF
C------ REAL PART OF POLYNOMIAL
      DO 20 I=0,MAXDG
         IF(MOD(I,4).EQ.0) THEN
            COEFBS(I)=COEFBS(I)+COEFG(I)
         ELSEIF(MOD(I,4).EQ.2) THEN
            COEFBS(I)=COEFBS(I)-COEFG(I)
         ENDIF
   20 CONTINUE
C------ IMAGINARY PART OF POLYNOMIAL
      DO 30 I=0,MAXDU
         IF(MOD(I,4).EQ.1) THEN
            COEFBS(I)=COEFBS(I)+COEFU(I)
         ELSEIF(MOD(I,4).EQ.3) THEN
            COEFBS(I)=COEFBS(I)-COEFU(I)
         ENDIF
   30 CONTINUE
C
      MAXBS=MAXDG
      IF(MAXBS.LT.MAXDU) MAXBS=MAXDU
C
      RETURN
      END
```

5 . GETBGU
次数を与えて伝達関数の実部・虚部の多項式をそれぞれ求める.
引数　NN : 次数（入力）
COEFG(N) : 実部の多項式の係数, Nは次数（≧0）（出力）
MAXDG : 実部の多項式の最大次数（出力）
COEFU(N) : 虚部の多項式の係数, Nは次数（≧0）（出力）
MAXDU : 虚部の多項式の最大次数（出力）
IERR : ≠0のときエラー（出力）

```
C**********************************************
C
      SUBROUTINE GETBGU(NN,COEFG,MAXDG,COEFU,MAXDU,IERR)
C
C       GET BESSEL FILTER POLYNOMIAL ( SEPARATLY ABOUT G,U )
C
C        NN   :  ORDER OF POLYNOMIAL
C        COEFG : COEFFICIENT OF POLYNOMIAL OF REAL PART      ( G )
C        MAXDG : MAXIMUM ORDER OF COEFG
C        COEFU : COEFFICIENT OF POLYNOMIAL OF IMAGINARY PART ( U )
C        MAXDU : MAXIMUM ORDER OF COEFU
C        IERR  : .NE.0 ==> ERROR
C
C**********************************************
C
      DIMENSION COEFG(0:50),COEFU(0:50)
      REAL*8    DCOEFG(0:50),DCOEFU(0:50)
C
C  MRKEXG   MARK FOR NEXT EXTENSION OF POLYNOMIAL ORDER  ( REAL PART )
C  MRKEXU                                                ( IM. PART )
C  WCOEFG    COEFFICIENT OF EACH TERM    ( REAL PART )
C  WCOEFU                                ( IMAGINARY PART )
C  IDEGG    ORDER OF EACH TERM         ( FOR REAL PART )
C  IDEGU                                ( FOR IMAGINARY PART )
C
      DIMENSION MRKEXG(1000),MRKEXU(1000),IDEGG(1000),IDEGU(1000)
      REAL*8    WCOEFG(1000),WCOEFU(1000)
C
C====== CLEAR OUTPUT BUFFER =========================
      DO 10 I=0,50
         COEFG(I)=0.0
         COEFU(I)=0.0
         DCOEFG(I)=0.0D0
```

```
         DCOEFU(I)=0.0D0
   10 CONTINUE
C
      IERR=0
C----- NN ILLEGAL ? -------------------------
      IF(NN.LE.0.OR.NN.GT.20) THEN
         IERR=1
         WRITE(6,'(/4X,40(''#''),''    GETBGU ERROR NN='',I8,3X,
     +      40(''#''))') NN
         RETURN
C----- NN=1 ---------------------------------
      ELSEIF(NN.EQ.1) THEN
         COEFG(0)=1.0
         COEFU(1)=1.0
         MAXDG=0
         MAXDU=1
         RETURN
      ENDIF
C
C====== CLEAR WORK BUFFER ============================
      DO 20 I=1,1000
         MRKEXG(I)=0
         MRKEXU(I)=0
         WCOEFG(I)=0.0D0
         WCOEFU(I)=0.0D0
         IDEGG(I)=0
         IDEGU(I)=0
   20 CONTINUE
      MRKEXG(1)=1
      WCOEFG(1)=1.0D0
      IDEGG(1)=0
      WCOEFU(1)=1.0D0
      IDEGU(1)=1
C..... NUMBER OF TERM
      NTERMG=1
      NTERMU=1
C
C=========================================
C  CALCULATE EACH TERM COEFFICIENT
C=========================================
      DO 30 I=2,NN
C
C===== REAL PART ============================
         NG=NTERMG
         DO 40 J=1,NTERMG
C------ NO EXTENSION TERM ---------------------------
            IF(MRKEXG(J).EQ.0) THEN
               MRKEXG(J)=1
               WCOEFG(J)=WCOEFG(J)*DBLE(2*I-1)
C------ EXTENSION TERM
            ELSE
C     ( CREATE NEW TERM )
               NG=NG+1
               WCOEFG(NG)=-WCOEFG(J)/DBLE(2*I-3)
               IDEGG(NG)=IDEGG(J)+2
               MRKEXG(NG)=0
C
               WCOEFG(J)=WCOEFG(J)*DBLE(2*I-1)
               MRKEXG(J)=1
            ENDIF
   40    CONTINUE
         NTERMG=NG
C
C======= IMAGINARY PART ============================
         NU=NTERMU
         DO 50 J=1,NTERMU
C------ NO EXTENSION TERM --------------------------
            IF(MRKEXU(J).EQ.0) THEN
               MRKEXU(J)=1
               WCOEFU(J)=WCOEFU(J)*DBLE(2*I-1)
C------ EXTENSION TERM
            ELSE
C     ( CREATE NEW TERM )
               NU=NU+1
               WCOEFU(NU)=-WCOEFU(J)/DBLE(2*I-3)
               IDEGU(NU)=IDEGU(J)+2
               MRKEXU(NU)=0
C
               WCOEFU(J)=WCOEFU(J)*DBLE(2*I-1)
               MRKEXU(J)=1
            ENDIF
   50    CONTINUE
         NTERMU=NU
C
   30 CONTINUE
C
C==============================================
C     GROUPING EACH ORDER
C==============================================
C----- REAL PART --------------------------
      MAXDG=0
      DO 60 I=1,NTERMG
         IF(IDEGG(I).GT.MAXDG) MAXDG=IDEGG(I)
   60    DCOEFG(IDEGG(I))=DCOEFG(IDEGG(I))+WCOEFG(I)
      DO 65 I=0,MAXDG
   65    COEFG(I)=DCOEFG(I)
C----- IMAGINARY PART -------------------------
      MAXDU=0
      DO 70 I=1,NTERMU
         IF(IDEGU(I).GT.MAXDU) MAXDU=IDEGU(I)
   70    DCOEFU(IDEGU(I))=DCOEFU(IDEGU(I))+WCOEFU(I)
C--- REAL*8 ==> REAL*4
      DO 75 I=0,MAXDU
   75    COEFU(I)=DCOEFU(I)
C
      RETURN
      END
```


29

6 . GETBTX

伝達関数の多項式が与えられた値をとるときの入力値xを求める.

引数　AP：多項式の値に関するパラメータ

COEFFB(N)：多項式の係数（入力）

MAXDBS：多項式における最大次数（入力）

TX：次の式を満足する値（出力）

$$\frac{1}{(1+(A_P)^2)^{1/2}} = \left|\sum_{n=1}^{MAXDBS} C_{OEFFB}(n)\cdot(iT_X)^n\right|$$

IERR：≠0のときエラー（出力）

```
C****************************************************
C
      SUBROUTINE DETBTX(AP,COEFFB,MAXDBS,TX,IERR)
C
C      DETERMINE BESSEL FILTER POLYNOMIAL X VALUE FOR GIVEN AP
C
C ( I ) AP     : PASS BAND MAXIMUM ATTENATION
C ( I ) COEFFB : COEFFICIENTS OF POLYNOMIAL
C ( I ) MAXDBS : MAX. ORDER OF THE POLYNOMIAL
C ( O ) TX     : THE VALUE WHICH GIVES AP IN BESSEL POLYNOMIAL
C ( O ) IERR   : ERROR FLAG
C
C****************************************************
C
      DIMENSION COEFBS(0:50),COEFFB(0:50)
C
C----- ATTENATION VALUE
      BP=SQRT(1.0+AP*AP)
C
C----- NORMALIZING OF POLYNOMIAL
         WW=COEFFB(0)
         DO 20 J=0,MAXDBS
            COEFBS(J)=COEFFB(J)/WW
   20    CONTINUE
C----- GET TX BY THE BISECTION METHOD
         TX1=0.0
         TX2=1.0
         DO 30 J=1,50
            CALL VALBPL(TX2,COEFBS(0),MAXDBS,VAL)
            IF(VAL-BP.GT.0.0) GOTO 40
            TX2=TX2*2.0
   30    CONTINUE
   40    CONTINUE
C
         DO 50 J=1,150
            TX=(TX1+TX2)/2.0
            CALL VALBPL(TX,COEFBS(0),MAXDBS,VAL)
            IF(VAL-BP.GT.0.0) THEN
               TX2=TX
            ELSE
               TX1=TX
            ENDIF
            IF(ABS(TX2-TX1)/ABS(TX2+TX1).LT.1.0E-5) GOTO 60
   50    CONTINUE
         WRITE(6,'('' GETBSN ERROR  NO CONVERGENCE OF X'')')
         GOTO 100
C
   60    CONTINUE
C
      IERR=0
      RETURN
C---- ERROR
  100 CONTINUE
      IERR=1
      RETURN
      END
```

7 . VALBPL

伝達関数の多項式について入力値xが与えられたときの出力を求める.

引数　X：入力値（入力）

COEFF(I)：伝達関数の多項式の各係数（入力）

N：多項式における最大次数（入力）

VAL：以下の式に示す値を与える.（出力）

$$V_{AL} = \left|\sum_{n=1}^{N} C_{OEFF}(n)\cdot(iX)^n\right|$$

```
C****************************************************
C
      SUBROUTINE VALBPL(X,COEFF,N,VAL)
C
C       GET OUTPUT VALUE FOR BESSEL FILTER POLYNOMIAL FUNCTION
C
C         X     : GIVEN VALUE FOR CALCURATION
C         COEFF : COEFFICIENTS OF POLYNOMIAL
C         N     : THE ORDER OF THE POLYNOMIAL
C         VAL   : OUTPUT
C                 VAL=ABS( SIGMA (COEFF(J)*(IX)**J) )
C
C****************************************************
C
      DIMENSION COEFF(0:N)
C
      REAL*8 VR,VI,XN
C----- INITIALIZE
      VR=0.0D0
      VI=0.0D0
      XN=1.0D0
C
      DO 10 I=0,N
C.... REAL PART
         IF(MOD(I,4).EQ.0) THEN
            VR=VR+DBLE(COEFF(I))*XN
         ELSEIF(MOD(I,4).EQ.2) THEN
            VR=VR-DBLE(COEFF(I))*XN
C..... IMAGINARY PART
         ELSEIF(MOD(I,4).EQ.1) THEN
```

```
            VI=VI+DBLE(COEFF(I))*XN
         ELSEIF(MOD(I,4).EQ.3) THEN
            VI=VI-DBLE(COEFF(I))*XN
         ELSE
            WRITE(6,'(''  VALBPL  ERROR I='',I5)') I
         ENDIF
C......  XN=X**(I)
         XN=XN*DBLE(X)
   10 CONTINUE
C
      VAL=DSQRT(VR*VR+VI*VI)
C
      RETURN
      END
```

8．BAIHIT

ベアストウ-ヒチコック法により，与えられた代数方程式の根を求める．

引数　N：方程式の最大次数（入力）

A(I)：方程式の各項の係数（N-Iが次数）（入力）

Z：方程式の複素根（出力）

```
C************************************************
C
      SUBROUTINE BAIHIT(N,A,Z)
C
C       CALCULATE ROOTS OF POLYNOMIAL BY BAIRSTOW-HITCHICOCK METHOD
C         ( 1989/5/12 PROGRAMMED BY KATSUMATA )
C        ( AFTER 'SUUCHI-KENSAN-HOU' BY HAYATO TOGAWA
C          CORONA PUBLISHING CO., LTD.   1981 )
C
C          N : ORDER OF POLYNOMIAL
C          A(I) : COEFFIENTS OF POLYNOMIAL
C                   A(I) COEFFICIENT FOR N-I POWERS OF X
C                   COEFFIENT OF N POWERS OF X IS 1.0
C          Z(I) : COMPLEX ROOTS (I : 1 -> N )
C
C************************************************
C
      DIMENSION A(N)
      COMPLEX Z(N)
C P,Q : FACTOR POLYNOMIAL
C             PBUR(I):  COEFFICIENT OF 1 POWER OF X
C             QBUF(I):  COEFFICIENT OF 0 POWER OF X
C             I: INDEX FOR POLYNOMIAL
      REAL*8    PBUF(20),QBUF(20)
      REAL*8    AWRK(50),B(50),BWRK(50),C(50)
     +         ,P,Q,BN,BNM1
     +         ,CNM1,CNM2,CNM3,A11,A12,A21,A22,B1,B2,DP,DQ,H
     +         ,RA,SA,RB,SB
C
C------- PRE-CHECK FOR POLYNIMIAL DIVIDE
      IF(N.EQ.1) THEN
         Z(1)=CMPLX(-A(1),0.0)
         RETURN
      ELSEIF(N.EQ.2) THEN
         NPQ=1
         PBUF(NPQ)=A(1)
         QBUF(NPQ)=A(2)
         GOTO 20
      ELSEIF(N.LT.1) THEN
         WRITE(6,'(''  BAIHIT  ORDER ERROR N='',I3)') N
         RETURN
      ENDIF
C
C------ NUMBER OF POLYNOMIALS WITH 2 POWERS
      NPQ=0
C------ ORDER OF REMAINED POLYNOMIAL
      NA=N
C
      DO 40 I=1,N
         AWRK(I)=A(I)
   40 CONTINUE
C
   10 CONTINUE
C=======================================================
C     GET P & Q ( POLYNOMIAL COEFFICIENT OF 2 POWERS )
C             BY ITERATION METHOD
C=======================================================
   30 CONTINUE
C------ INITIAL SETTING OF P & Q
         IF(ABS(AWRK(NA)).GT.1.0E-2) THEN
            P=(ABS(AWRK(NA)))**(1.0/REAL(NA))
            Q=P
            RJUGDE=P
         ELSE
            P=1.0D0
            Q=1.0D0
            RJUGDE=P
         ENDIF
C
         DLIMIT=1.0E-12
  100    CONTINUE
         DO 50 KKK=1,50
C------- DIVIDE BY (X**2+P*X+Q)
            CALL DDVPLN(AWRK(1),NA,B(1),P,Q,RA,SA)
C....... END?
            IF(ABS(RA)/RJUGDE.LT.DLIMIT.AND.ABS(SA)/RJUGDE.LT.DLIMIT)
     +                        THEN
               NA=NA-2
               NPQ=NPQ+1
               PBUF(NPQ)=P
               QBUF(NPQ)=Q
C
               IF(NA.GE.3) THEN
                  DO 80 I=1,NA
                     AWRK(I)=B(I)
   80             CONTINUE
```

```
               GOTO 30
            ELSEIF(NA.EQ.2) THEN
               NPQ=NPQ+1
               PBUF(NPQ)=B(1)
               QBUF(NPQ)=B(2)
               GOTO 20
            ELSEIF(NA.EQ.1) THEN
               Z(N)=DCMPLX(-B(1),0.0D0)
               GOTO 20
            ENDIF
         ENDIF
C====== GET NEXT P & Q VALUE ============================
         NB=NA-2
         DO 70 I=1,NB
   70       BWRK(I)=B(I)
         BWRK(NB+1)=0.0
C
         CALL DDVPLN(BWRK(1),NB+1,C(1),P,Q,RB,SB)
C
         BN=B(NB)
         IF(NB.EQ.1) THEN
            BNM1=1.0
         ELSE
            BNM1=B(NB-1)
         ENDIF
C
         IF(NB.EQ.1) THEN
            CNM1=1.0
            CNM2=0.0
            CNM3=0.0
         ELSEIF(NB.EQ.2) THEN
            CNM1=C(NB-1)
            CNM2=1.0
            CNM3=0.0
         ELSEIF(NB.EQ.3) THEN
            CNM1=C(NB-1)
            CNM2=C(NB-2)
            CNM3=1.0
         ELSE
            CNM1=C(NB-1)
            CNM2=C(NB-2)
            CNM3=C(NB-3)
         ENDIF
C
         A11=BN-CNM2*Q-CNM1*P
         A12=BNM1-CNM3*Q-CNM2*P
         A21=-CNM1*Q
         A22=BN-CNM2*Q
         B1=RA
         B2=SA
C
         DP=(B1*A22-A12*B2)/(A11*A22-A12*A21)
         DQ=(A11*B2-A21*B1)/(A11*A22-A12*A21)
         P=P+DP
         Q=Q+DQ
```



```
C------- GOTO NEXT ITERATION
   50    CONTINUE
         DLIMIT=DLIMIT*2.0
         GOTO 100
C
C======================================================
C     CALCULATE ROOTS
C======================================================
   20 CONTINUE
      DO 60 I=1,NPQ
         H=PBUF(I)*PBUF(I)-4.0*QBUF(I)
         IF(H.GE.0.0) THEN
            Z(2*I-1)=DCMPLX((-PBUF(I)+DSQRT(H))/2.0,0.0D0)
            Z(2*I)  =DCMPLX((-PBUF(I)-DSQRT(H))/2.0,0.0D0)
         ELSE
            Z(2*I-1)=DCMPLX(-PBUF(I)/2.0, DSQRT(-H)/2.0D0)
            Z(2*I)  =DCMPLX(-PBUF(I)/2.0,-DSQRT(-H)/2.0D0)
         ENDIF
   60 CONTINUE
C
      RETURN
      END
```

9．DDVPLN

多項式を与えられた２次式により除する．

引数　A(I)：除算の対象となる多項式（入力）
　　　　　（A(I)はN-I次の項の係数）
　　N：A(I)の最大次数（入力）
　　B(I)：除算の答の多項式（出力）
　　　　　（B(I)はN-2-I次の項の係数）
　　P,Q：A(I)を除する２次式（X^2+PX+Q）の係数
　　　　　（入力）
　　R,S：余りの式（RX+S）（出力）

```
C*************************************************
C
      SUBROUTINE DDVPLN(A,N,B,P,Q,R,S)
C
C     DIVIDE POLYNOMIAL BY (X*X+PX+Q)
C      (A*X) = (X*X+P*X+Q)*(B*X) + (RX+S)
C       ( A*X = X**N + A1*X**(N-1) + ...... + AN )
C
C*************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(N),B(N)
C
      IF(N.EQ.1) THEN
         R=1.0
         S=A(1)
      ELSEIF(N.EQ.2) THEN
         R=A(1)-P
         S=A(2)-Q
      ELSE
```

```
      DO 10 I=1,N-2
         IF(I.EQ.1) THEN
            B(I)=A(1)-P
         ELSEIF(I.EQ.2) THEN
            B(I)=A(2)-P*B(1)-Q
         ELSE
            B(I)=A(I)-P*B(I-1)-Q*B(I-2)
         ENDIF
   10 CONTINUE
C------ AMARI -----------------------------
      IF(N.EQ.3) THEN
         R=A(2)-Q-P*B(1)
         S=A(3)-Q*B(1)
      ELSE
         R=A(N-1)-Q*B(N-3)-P*B(N-2)
         S=A(N)-Q*B(N-2)
      ENDIF
    ENDIF
C
    RETURN
    END
```

10. TANDEM（斎藤(1978)）による．

複数の基本フィルタの処理を行う．

引数　X(I):入力データ

Y(I):出力データ

N:データ数（入力）

H(I):フィルタの係数（入力）

M:処理する基本フィルタの数（入力）

GN:倍率に関する係数（入力）

z変換は次の式による．

$$F(z)=G_N\prod_{i=1}^{M}\frac{1+H_{(4i-3)}z+H_{(4i-2)}z^2}{1+H_{(4i-1)}z+H_{(4i)}z^2}$$

NML:フィルタ処理を行う方向（入力）

>0：通常の処理

<0：時間について逆方向に処理する．

```
C==========================================================
      SUBROUTINE TANDEM(X,Y,N,H,M,GN,NML)
C
C      RECURSIVE FILTERING IN SERIES
C
C ( I ) X : INPUT TIME SERIES
C ( O ) Y : OUTPUT TIME SERIES (MAY BE EQUIVALENT TO X )
C ( I ) N : LENGTH OF X & Y
C ( I ) H : FILTER COEFFICIENT
C ( I ) M : ORDER OF FILTER
C ( I ) GN: GAIN FACTOR
C ( I ) NML : >0 : FOR NORMAL DIRECTION FILTERING
C             <0 : FOR REVERSE DIRECTION FILTERING
C
C=======================================================
      DIMENSION X(N),Y(N),H(*)
      IF(N.LE.0.OR.M.LE.0) GO TO 2
      CALL RECFIL(X(1),Y(1),N,H(1),NML)
      IF(M.GT.1) THEN
C*** 2-ND AND AFTER
         DO 1 I=2,M
            CALL RECFIL(Y(1),Y(1),N,H(I*4-3),NML)
    1    CONTINUE
      ENDIF
C------ MULTIPLY GAIN FACTOR -------------------------
      DO 20 I=1,N
         Y(I)=GN*Y(I)
   20 CONTINUE
C
      RETURN
C*** ERROR
    2 WRITE(6,3) N,M
    3 FORMAT(//1X,5('?'),3X,'(TANDEM)',3X,'INVALID INPUT',3X,'N=',I5,
     $ 3X,'M=',I5,5('?')//)
      RETURN
      END
```

11. RECFIL（斎藤(1978)）による．

基本フィルタの処理を行う

引数　X(I):入力データ

Y(I):出力データ

N:データ数（入力）

H(I):フィルタの係数（入力）

z変換は次の式による．

$$F(z)=\frac{1+H_{(1)}z+H_{(2)}z^2}{1+H_{(3)}z+H_{(4)}z^2}$$

NML:フィルタ処理を行う方向（入力）

>0：通常の処理

<0：時間について逆方向に処理する．

```
C=====================================================
      SUBROUTINE RECFIL(X,Y,N,H,NML)
C
C      RECURSIVE FILTERING
C    X : INPUT TIME SERIES
C    Y : OUTPUT TIME SERIES (MAY BE EQUIVALENT TO X )
C    N : LENGTH OF X & Y
C    H : FILTER COEFFICIENT
C    NML : >0 : FOR NORMAL DIRECTION FILTERING
C          <0 : FOR REVERSE DIRECTION FILTERING
C
C======================================================
      DIMENSION X(N),Y(N),H(*)
      IF(N.LE.0) GO TO 4
      IF(NML.GE.0) then
C*** NORMAL FILTERING
         J=1
         JD=1
      else
C*** REVERSE FILTERING
         J=N
```

```
         JD=-1
      endif
C
    2 A=H(1)
      AA=H(2)
      B=H(3)
      BB=H(4)
      U1=0.
      U2=0.
      V1=0.
      V2=0.
C*** FILTERING
      DO 3 I=1,N
         U3=U2
         U2=U1
         U1=X(J)
         V3=V2
         V2=V1
         V1=U1+A*U2+AA*U3-B*V2-BB*V3
         Y(J)=V1
         J=J+JD
    3 CONTINUE
C*** EXIT
      RETURN
C*** ERROR
    4 WRITE(6,5) N
    5 FORMAT(//1X,5('?'),3X,'(RECFIL)',3X,'INVALID INPUT',3X,'N=',I5,
     $ 3X,5('?')//)
      RETURN
      END
```