電子回路論 第 15 回 (最終回)

勝本信吾
東京大学理学部・理学系研究科 (物性研究所)

2015 年 1 月 21 日

前回のサマリー

ディジタル回路　順序回路 (フリップフロップ，カウンタ)
回路の簡単化　カルノー図，クワイン・マクラスキー法
ディジタル-アナログ変換　抵抗ストリング型，ラダー型

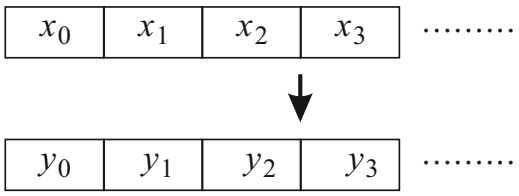
6.6　ディジタルフィルター

一旦ディジタルに落とした信号は，回路中で数値として扱うことができるので非常に自由度が高く，安定な処理が可能である．このような「処理」の内，ディジタル信号からディジタル信号への変換処理を**フィルター** (filter) 作用と呼ぶ．ディジタル信号処理 (digital signal processing, DSP) も極めて巨大な分野であり，講義の 1 回程度ではほとんど「舐めてみる」程度のことしかできないが，ディジタルフィルターの初歩について考え，入り口を経験することにしよう．なお，この **DSP** という略語は「ディジタル信号処理」という意味，および「ディジタル信号プロセッサー」(digital signal processor) の両方に用いられ，混用されている．

ディジタルフィルターは物理的に行っていることはアナログフィルターとは全くと言って良い程異なっているが，(5.89) 式のように応答関数の形に書くと，類似の舞台の上で考えることができる対象であることがわかる．ディジタルフィルターは，アナログフィルターがスマートに滑らかに行っていたことをゴツゴツと各段階計算で押し切っているので必要な回路リソースは桁違い以上に大きいが，特性が部品特性に制約されず設計に対する忠実度が高く，自由な設計が可能であり，何より，制御信号などによって容易に特性を変化させることができる．

6.6.1　ブロック・ダイアグラム

ディジタルフィルターの働きを簡単にまとめると，右図のように見るのが 1 つの考え方であろう．すなわち，時系列で並んだ数列 $\{x_i\} = (x_0, x_1, \cdots)$ に対して，別の数列 $\{y_i\} = (y_0, y_1, \cdots)$ を得るプロセスである．ここで，数列値 y_n は，x_n およびこれよりも以前の k 個の x_i の値によって決定されるとすると，$k+1$ 変数の関数 F を使って



$$y_n = F(x_{n-k}, x_{n-k+1}, \cdots, x_n) \tag{6.12}$$

と書けることになる．

F として現実には数列要素間および定数との間の加減乗除がはいることがほとんどである．これまでも回路を機能のまとまりで記述した**ブロック・ダイアグラム** (block diagram) で表すことを良く行ってきたが，ディジタルフィルターの F は，簡単な要素を使ってブロック・ダイアグラムで表すことが良く行われる．その基本要素とは図 6.20 の

3 つであり，(a) はディジタル回路のバッファの形をしているが，記入されている数倍だけする，**定数倍回路**，である．(b) は信号の和を取る**加算回路**である．(c) は，信号を時間領域で 1 単位だけ遅らせる**遅延回路**である．

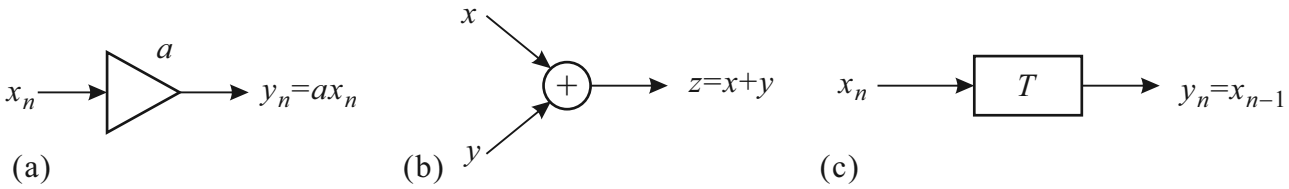


図 6.20　ディジタルフィルター用ブロックダイアグラムに使われる，「基本ブロック」．(a) 定数倍回路．(b) 加算回路．(c) 遅延回路 (シフトレジスタ)．

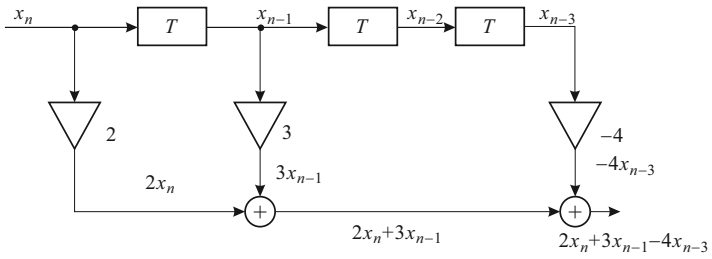


図 6.21　式 (6.13) のダイアグラム化．

例として

$$y_n = 2x_n + 3x_{n-1} - 4x_{n-3} \tag{6.13}$$

をダイアグラム化すると左の図のようになる．

6.6.2　z 変換空間表示

ディジタルの線形応答回路においてもアナログ回路同様, z 変換空間において伝達関数を使って応答関係を表すことができることを示した．そこで，ディジタルフィルターも (片側)z 変換空間で表すことを考える．

ここではやはり例として (6.13) を考える．今の場合，フィルターの入出力の z 変換は定義 (5.83) より，

$$X(z) = \sum_{n=0}^{\infty} x_n z^{-n}, \quad Y(z) = \sum_{n=0}^{\infty} y_n z^{-n} \tag{6.14}$$

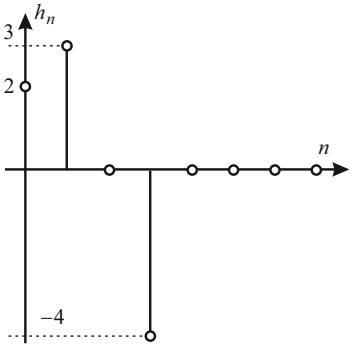
である．ここで，表 5.2 に示した片側 z 変換の性質を思い出すと，線形性より加算や定数倍はそのまま記述することができ，周期 τ の k 回文の遅延は時間移動で z^{-k} をかければ良いことがわかる．以上より，z 変換空間では

$$Y(z) = 2X(z) + 3z^{-1}X(z) - 4z^{-3}X(z) = (2 + 3z^{-1} - 4z^{-3})X(z) \tag{6.15}$$

で表されることがわかる．これより，伝達関数を $H(z)$ とすると，(5.89) より

$$H(z) = 2 + 3z^{-1} - 4z^{-3} \tag{6.16}$$

となることがわかる．



伝達関数は $t = 0$ でインパルス入力があった時の応答の z 変換であるから，これより，このディジタルフィルターにインパルス入力があると，図のような応答があることになる．一般にフィルターの関数 F(6.12) が,

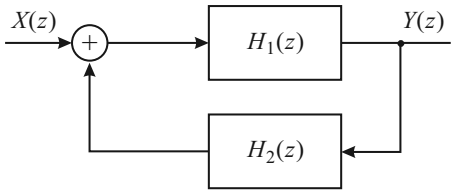
$$F(x_{n-k}, x_{n-k+1}, \cdots, x_n) = \sum_{j=0}^{k} a_j x_{n-j} \tag{6.17}$$

のようにかける場合，伝達関数は

$$H(z) = \sum_{j=0}^{k} a_i z^{-j} \tag{6.18}$$

となる．

6.6.3　ブロックダイアグラムと伝達関数



z 変換空間内でもブロックダイアグラムを考えることができる．この際，伝達関数をブロックで表してしまうのが簡単である．左図のように，2 つの伝達関数 $H_1(z)$ と $H_2(z)$ とを使って，フィードバック構造をつくるブロックダイアグラムを考えてみる．$X(z)$ から + 演算を経て H_1 へ入力される関数を $W(z)$ と置くと，

$$Y(z) = H_1(z)W(z) = H_1(z)(X(z) + H_2(z)Y(z)),$$
$$\therefore Y(z) = \frac{H_1(z)}{1 - H_1(z)H_2(z)}X(z).$$

すなわち，全体の伝達関数は

$$H(z) = \frac{H_1(z)}{1 - H_1(z)H_2(z)} \tag{6.19}$$

これは，むろん，アナログ回路の場合の (3.7) 式に対応するもので，フィードバック回路の場合，一般に

$$(\text{全体の伝達関数}) = \frac{(\text{直接部分の伝達ゲイン})}{1 - (\text{フィードバック部分の伝達ゲイン})} \tag{6.20}$$

となることを示している．ここで，フィードバックをかけると，式 (6.12) の引数が形式上無限に続くことになることに注意しよう．

6.6.4 ディジタルフィルターの分類

ディジタルフィルターは，上のようにフィードバックループを持つために (6.12) の引数が無限に続く形になる無限インパルス応答 (infinite impulse response, IIR) フィルターと，有限長の引数で表すことができる有限インパルス応答 (finite impulse response, FIR) フィルターとに分類することができる．フィードバックループがあってもインパルス応答が有限になる場合 (周波数サンプリングフィルタなど) は FIR フィルターである．

また，通過周波数帯域により，アナログの場合と同様，低域，高域，バンド (Low, High, Band) 通過フィルター (それぞれ，LPF，HPF，BPF)，また帯域除去フィルター (band elimination filter, BEF) などにも分類される．振幅特性形状からは，これもアナログ同様，バタワース，チェビシェフ，などの分類が可能である．

6.6.5 FIR フィルター

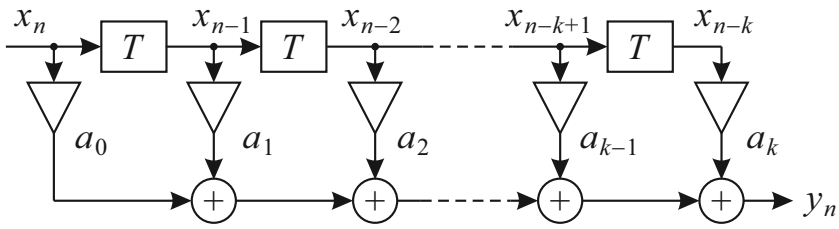


図 6.22 FIR フィルターの一般形 (6.17) のブロックダイアグラム．

線形 FIR フィルターは，一般に (6.17) の関数で表され，図 6.22 のようなブロックダイアグラムが描ける．加算数列が有限であるから，伝達関数 (6.18) には極がなく安定である．FIR フィルターの周波数特性は，(6.18) より，$z = e^{i\omega\tau}$ とおいて，

$$H(e^{i\omega\tau}) = \sum_{j=0}^{k} a_j e^{-ij\omega\tau} \tag{6.21}$$

である．

まず，もっとも単純な移動平均，および隣接差分，

$$F_{\pm}(x_n, x_{n-1}) = (x_n \pm x_{n-1})/2 \tag{6.22}$$

については,

$$H_{\pm}(e^{i\omega\tau}) = e^{-i\omega\tau/2}\begin{pmatrix}\cos(\omega\tau/2)\\ i\sin(\omega\tau/2)\end{pmatrix} \tag{6.23}$$

となる.

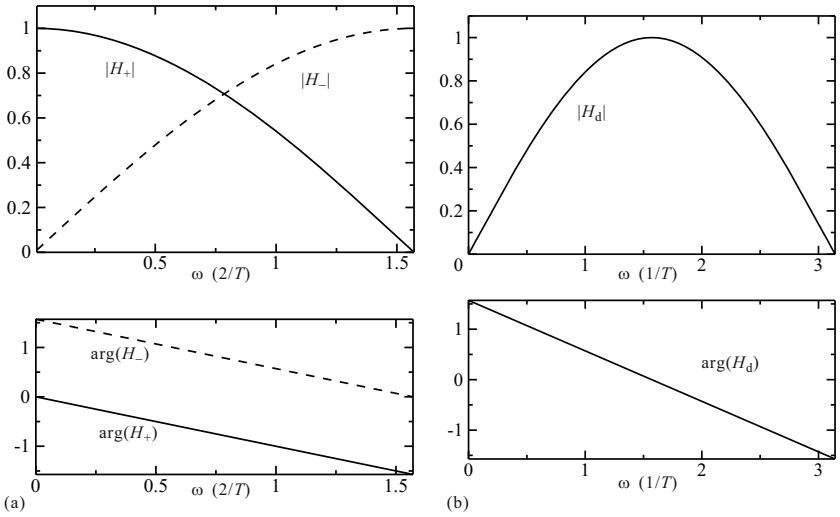


図 6.23　(a) 移動平均，隣接差分操作によるフィルターの周波数応答 (上が伝達関数の絶対値，下が位相) 特性．ω は $2/\tau$ 単位で計っているので，$\pi/2$ 以下の領域に限られる．(b) 移動平均の差分に対して (a) と同様の周波数特性．ただし，ω は $1/\tau$ 目盛り．

ω の区間 $[0, \pi/\tau]$ に対する周波数特性は，左の図のように移動平均 H_+ がローパスフィルター特性，隣接差分 H_- がハイパスフィルター特性となる．また位相は下の図のように ω に対してリニアに変化する．これは FIR フィルター一般に成立する特徴である．

移動平均の差分

$$F_{\mathrm{d}} = [(x_n + x_{n-1}) - (x_{n-1} + x_{n-2})]/2 = [x_n - x_{n-2}]/2 \tag{6.24}$$

は,

$$H_{\mathrm{d}} = (1 - e^{-2i\omega\tau})/2 = ie^{-i\omega\tau}\sin\omega\tau \tag{6.25}$$

より，図 6.23(b) のようにバンドパスフィルターとして働く．なお，これらディジタルフィルターの有効周波数範囲は，標本化定理より $1/2T$ 以下である．

6.6.6　IIR フィルター

本節冒頭で「ディジタルフィルターは安定」と述べたが FIR フィルターの場合であり，フィードバックによってインパルス応答が無限時間続く IIR フィルターの場合は安定性が問題となる．安定であるためにはインパルス応答を y_n として,

$$\lim_{n\to\infty} y_n = 0, \tag{6.26}$$

すなわち，無限に続くとしても 0 に収束することが必要である．

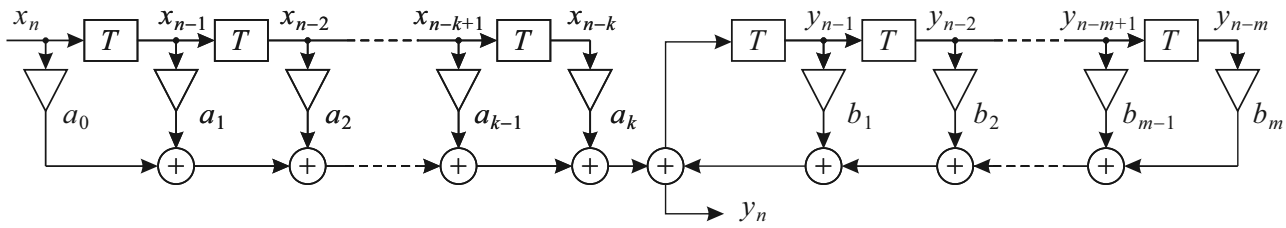


図 6.24　IIR フィルタの一般形ダイアグラム

IIR フィルタをダイアグラムで一般的に表すと，図 6.24 のように，FIR フィルタを通過した後，フィードバックループを経て出力するものが考えられる．関数形で表すと

$$y_n = \sum_{l=0}^{k} a_l x_{n-l} + \sum_{j=1}^{m} b_j y_{n-j} \tag{6.27}$$

のように書ける．この両辺を z 変換すると，

$$Y(z) = X(z)\sum_{l=0}^{k} a_l z^{-l} + Y(z)\sum_{j=1}^{m} b_j z^{-j} \tag{6.28}$$

であるから，伝達関数は

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\displaystyle\sum_{l=0}^{k} a_l z^{-l}}{1 - \displaystyle\sum_{j=1}^{m} b_j z^{-j}} \tag{6.29}$$

となる．

IIR フィルタは，ループを使用することで FIR フィルタに比べて少ない回路リソースで必要なフィルタ特性を得られる可能性がある一方，安定条件 (6.26) を満たさなければならない点が設計上の問題点となる．簡単な安定条件として，(6.29) の分母がゼロになる極となる z が，z 変換の収束域である $|z| > 1$ になければ良い．すなわち，極が単位円内に入っている必要がある．

6.6.7 ディジタルフィルターの設計

ディジタルフィルターの設計には様々な方法が存在する．FIR フィルタの場合，Ramez 法と呼ばれるものが良く使用され，Scilab などのプログラムでもデフォルトで ramez という関数が用意されている．が，ここでは原理・手続きが簡単な**窓関数法** (window function method) について簡単に紹介する．

周波数通過フィルターを考える場合，LPF について考えておけば十分 (他は周波数変換により LPF から変換できる) である．理想 LPF は，カットオフ周波数 ω_{c} 以下は完全透過し，以上を完全カットするもので，

$$G(e^{i\omega\tau}) = \begin{cases} 1, & |\omega| \le \omega_{\mathrm{c}}, \\ 0, & \omega_{\mathrm{c}} < |\omega| \le \omega_{\mathrm{N}} \end{cases} \tag{6.30}$$

と表される．$\omega_{\mathrm{N}} = \omega_{\mathrm{s}}/2 = \pi/\tau$ はナイキスト周波数である．(6.30) は 5.3 節で見たように，$[-\omega_{\mathrm{N}}, \omega_{\mathrm{N}}]$ の範囲で考えることにすればこの外側では $2\omega_{\mathrm{N}} = \omega_{\mathrm{s}}$ の周期を持つ周期関数としてフーリエ級数展開し，

$$G(e^{i\omega\tau}) = \frac{\omega_{\mathrm{c}}}{\omega_{\mathrm{N}}} \sum_{n=-\infty}^{\infty} \frac{1}{n\pi} \mathrm{sinc}\left(n\frac{\omega_{\mathrm{c}}}{\omega_{\mathrm{N}}}\right) e^{-ni\omega\tau} = \gamma_{\mathrm{c}} \sum_{n=-\infty}^{\infty} \mathrm{sinc}(n\gamma_{\mathrm{c}}) z^{-n} \tag{6.31}$$

が得られる．$\gamma \equiv \omega_{\mathrm{c}}/\gamma_{\mathrm{N}}$ である．すなわち，このような理想 LPF の n 次係数は，通過域幅のナイキスト周波数に対する比を γ として sinc 関数 $\mathrm{sinc}(n\gamma)$ で与えられることがわかる．

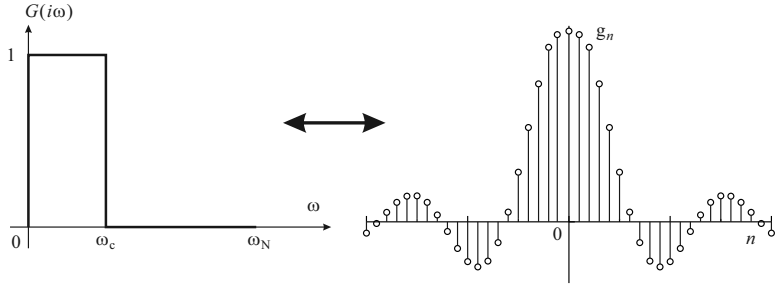
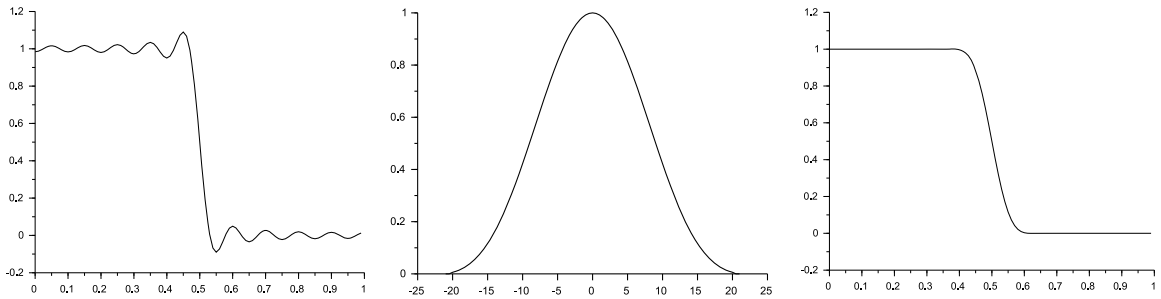


図 6.25　理想低域通過フィルタ (LPF) の周波数特性模式図 (左) と，これをフーリエ級数展開した際の n 次の項 $\mathrm{sinc}(n\omega_{\mathrm{c}}/\omega_{\mathrm{N}})$ の模式図 (右).



(a) (b) (c)

図 6.26 (a)(6.31) の理想 LPF の次数を $n = 20$ で打ち切ったフィルタの周波数特性．$\omega/\omega_{\mathrm{N}}$ を横軸にとった．また，$\omega_{\mathrm{c}}/\omega_{\mathrm{N}} = 0.5$ とした．(b) カイザー窓関数の例．$\alpha = 7$，$L = 20$ として (6.32) をプロットしたもの．(c) (a) のフィルタに，(b) の窓関数を乗じたフィルタの周波数特性．

FIR フィルタで理想 LPF を実現しようとすると，(6.31) のように無限次数のフィルタが必要となる．sinc 関数 $\mathrm{sinc}(x)$ は，$x \to \infty$ に対して $1/x$ で小さくなるので有限次数で打ち切ることを行うと，図 6.26(a) のように周波数特性が波を打つリップル特性となる．これは，sinc 関数中の sin 関数が有限振幅を持っているところで急にカットされたため，これに近い振動数の成分で滑らかにされることがなくなってリップルとして現れたものである．

そこで，急激に打ち切るのではなく何らかの**窓関数** (window function) を乗じて打ち切りの裾を緩やかにすることを考える．この目的のためによく使用されるのが，カイザー (Kaiser) 窓と呼ばれるもので

$$w_n = \begin{cases} \dfrac{I_0\left(\alpha\sqrt{1-(n/L)^2}\right)}{I_0(\alpha)} & |n| \le L, \\ 0 & |n| > L \end{cases} \tag{6.32}$$

で与えられる．ここで，I_0 は 0 次第 1 種変形ベッセル関数である．α，L はそれぞれリップルの最大値，遷移領域の幅 $\Delta\omega$ と関係するパラメタである．図 6.26(b) にその形状例を示した．これを (6.31) に乗じると，図 6.26(c) のような周波数特性となり，リップルがほぼ消えてバタワース (Butterworth) 特性に近づいていることがわかる (対数プロットをすると，リップルの存在は確認できる)．

IIR フィルターの場合は，伝達関数が有理式で書かれるため，要求に適合するアナログフィルターの伝達関数からの変形を行う形で設計されることが多い．バタワース特性を持つ n 次の LPF の伝達関数は

$$\Xi(s) = \sum_{k=0}^{N-1} \frac{\omega_k}{s - s_k}, \quad s_k = r_{\mathrm{c}} \exp\left[i\left\{\frac{\pi}{2} + \frac{(2k+1)\pi}{2n}\right\}\right] \tag{6.33}$$

と書ける．これは，インパルス応答のラプラス変換であるから，インパルス応答はヘビサイド関数を $u_{\mathrm{H}}(t)$ と書いて

$$\xi(t) = u_{\mathrm{H}}(t) \sum_{k=0}^{n-1} w_k \exp(s_k t) \tag{6.34}$$

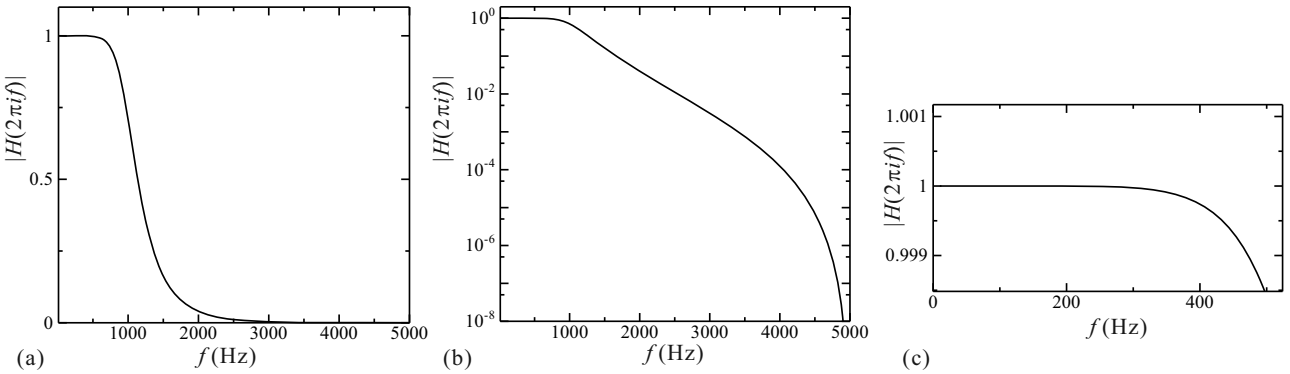
である．これを $\tau = 1$ で標本化 (離散化) すると，

$$h_n = h_{\mathrm{H}n} \sum_{k=0}^{n-1} w_k e^{n s_k}, \qquad \therefore H(z) = \sum_{k=0}^{n-1} \frac{w_k}{1 - \exp(s_k) z^{-1}}. \tag{6.35}$$

これは，(6.33) で

$$(s - s_k)^{-1} \to (1 - \exp(s_k) z^{-1})^{-1} \tag{6.36}$$

という置換を施した形である．(6.36) を用いて IIR フィルターに変換する方法を**インパルス不変法** (impulse invariant method) と呼ぶ．



coef.	0	1	2	3	4
a_n		2.3695130072	-2.3139884144	1.0546654059	$-1.8737949237\times10^{-1}$
b_n	$4.8243433577\times10^{-3}$	$1.9297373431\times10^{-2}$	$2.8946060146\times10^{-2}$	$1.9297373431\times10^{-2}$	$4.8243433577\times10^{-3}$

図 6.27　双一次 z 変換法で設計した IIR-LPF の周波数特性．バタワース特性について設計した．設計パラメタは本文を参照せよ．(a) 線形プロット．(b) 縦軸対数プロット．0 付近でもリップルが生じていない．(c) カットオフ周波数以下の特性の拡大図．リップルが見られない．下の表は，(6.38) の係数をサンプリング周波数 10kHz, 遮断周波数 1kHz に対して双一次 z 変換に対して計算したもの．

これに対して**双一次 z 変換法** (bilinear z-transform) では，

$$s \to \frac{1-z^{-1}}{1+z^{-1}} \tag{6.37}$$

という置換を行う．エイリアシングによる特性劣化が少ない方法として知られている．

図 6.27 はこのようにして，4 次のバタワース特性フィルタについて設計したもので伝達関数は

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3} - a_4 z^{-4}} \tag{6.38}$$

のように書け，係数は図 6.27 中のテーブルのようになる．図にも示したように，高々 4 次のフィルターでリップルが全くない特性を作ることができる．

6.6.8　複素信号処理と位相ロックループ

アナログ回路でも多数の信号線を使えば多次元データ伝送が可能であるが，ディジタルの場合は同期回路であれば多次元にすることに対する問題がより小さくなる．したがって，データ次元を 2 次元として複素数として扱うことも容易である．1 次元の実信号の ω 成分 $\cos(\omega t + \theta_0)$ に対して複素信号 $\exp[i(\omega t + \theta_0)]$ を作り出すには $\pi/2$ だけ位相の遅れた sin 成分を作り出して虚数単位 i を乗じて加えれば良いから，このプロセスをブロックダイアグラムで表すと，図 6.28(a) のようになる．ここで「ヒルベルト変換器」と書かれた部分が $\pi/2$ の位相遅れを生じる所であり，その伝達関数位相-周波数特性は図 6.28(b) のようになる．これも一種のフィルターであり，ヒルベルトフィルターと呼ばれる．

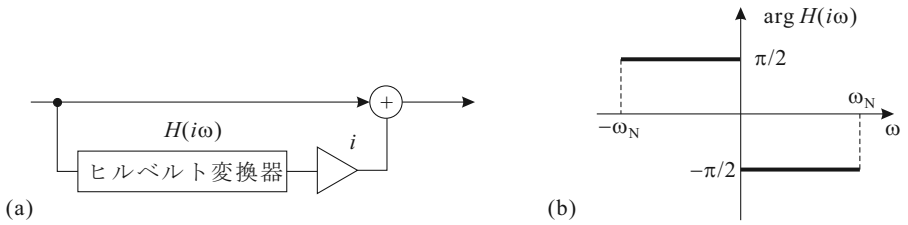


図 6.28　(a) ヒルベルト変換器による複素信号生成回路のブロックダイアグラム．(b) ヒルベルト変換器位相の ω 依存性．

これによって 2 次元化した信号を複素数として扱うことで様々な応用が可能となる．図 6.29 は位相ロックループ (phase-locked loop, PLL) と呼ばれる回路をブロックダイアグラムで模式的に表したものである．

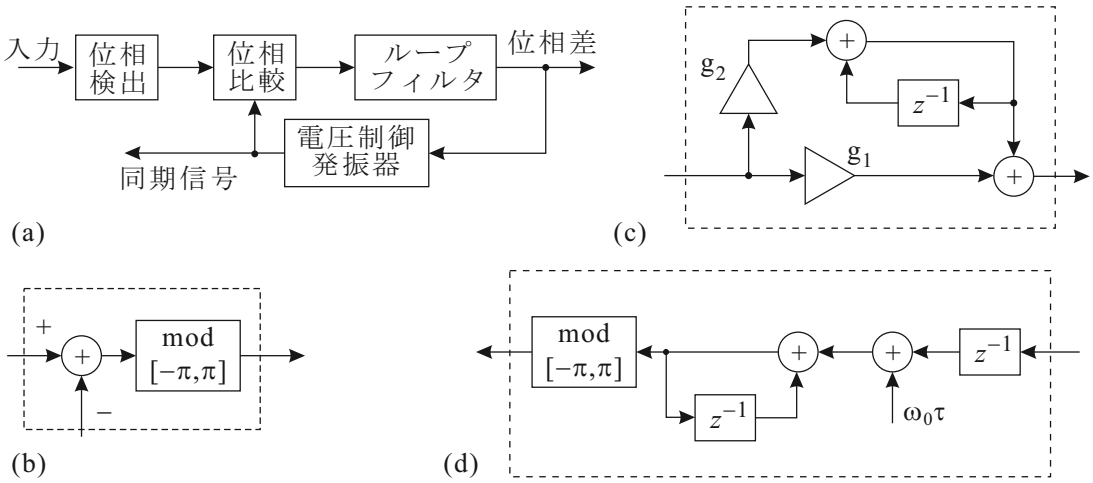


図 6.29　(a) 位相ロックループ回路のブロックダイアグラム．(b) ループフィルタ部分．(c) 位相比較部分．(d) 電圧制御発振器部分．

ここではディジタル回路を基本として説明するが，当然同機能の回路はアナログでも実現でき，PLL 機能全体が IC 化されて広く使われている．高周波で使用されることが多いため，ディジタル実装に関してはサンプリング周波数，演算速度等から一定の制限がある．一方，ディジタル回路の利点は，線形性が高く精度の良い回路が形成でき，応用の際にも様々な信号を自在に取り出せることにある．

図 6.29(a) に PLL 回路の全容を模式的に示した．PLL 回路に特徴的な部分は，電圧に応じて発振周波数が変化する電圧制御発振器 (voltage controlled oscillator, VCO) を備えている点である．VCO は，電圧入力 (今のがないときには適当な周波数 ω_0 で発振する PLL 回路ではまず，入力信号の位相を検出し，これを VCO の位相と比較し，位相差信号をループフィルタを通して出力する．これが VCO に入って発振周波数が変化し，全体のフィードバック回路により位相差信号が極小になるように調整される．入力信号と ω_0 が近ければ，VCO の発振周波数と位相は (一定さを除き) 入力信号にロックされる．以下各部に分けて回路構成を調べる．

位相検出・比較

まず，通常の実信号入力を図 6.28(a) の回路を通すことで複素信号化 $(x_n^{(\mathrm{r})}, x_n(\mathrm{i}))$ し，位相データ

$$\phi_n = \arctan \frac{x_n^{(\mathrm{i})}}{x_n^{(\mathrm{r})}} \tag{6.39}$$

を得る．同様のことを VCO の出力に対して行い，位相データ φ_n を得，図 6.29(b) の位相比較器では差分 $\phi_n - \varphi_n$ を計算する．常に $[-\pi, \pi]$ 内に値が来るように適当に 2π の加減算を行う．

ループ・フィルター

位相比較出力が送られるループ・フィルターは図 6.29(c) のような IIR フィルターで，伝達関数は

$$H(z) = \mathrm{g}_1 + \frac{\mathrm{g}_2}{1 - z^{-1}} \tag{6.40}$$

と書かれる．特に第 2 項の $(1 - z^{-1})^{-1}$ は 1 の z 変換であるから，インパルス入力に対して出力が一定で無限に続く，「完全積分型」と呼ばれる回路になっている．ここは，位相差から制御出力を作り出す部分で PLL 回路の特性を決める重要な部分である．位相差回路から誤差出力がなければループ・フィルターは一定の出力を出し続け，誤差出力に応じて常にこれをゼロにする方向に変化するから，「ロック」がかかった状態では VCO の出力周波数 ω_{v} は入力信号周波数 ω_{i} にロックされ，$\omega_{\mathrm{v}} = \omega_{\mathrm{i}}$ となる．(6.26) が満たされていないため，ループ・フィルターは必ずしも安定ではない．安定のためには条件

$$\mathrm{g}_2 < -2\mathrm{g}_1 + 4 \tag{6.41}$$

が満たされる必要がある．

VCO

図 6.29(d) に示した VCO 回路にはどこにも「発振部」がない．結局必要なのは位相情報であるため，位相情報だけを作っている．同期信号が必要な場合は位相から振動出力を得る回路を用意しこれに入力するようにする．

以上が PLL 回路であり，大変応用の広い回路である．信号選択，周波数逓倍，分割，変換などに使うことができるほか，VCO への制御出力を使って FM 検波などにも使える．

6.6.9 Δ-Σ 型 A-D コンバーター

Δ-Σ 型 AD 変換 (変調) は，オーバーサンプリング型 AD コンバーターの代表例で，1 ビットのコンパレーターを使い，量子化誤差をフィードバックすることで極めて誤差の大きな信号を「元手」として超高分解能ディジタル信号を作り出してしまう．実際には 2 次以上のフィルターを使用することが多いが，図 6.30(a) には，ごく簡単な原理を示してある．コンパレーターはあくまで 1 ビットであるが，出力を入力に戻して引き算をし，これを再びコンパレーターへ入力することを繰り返すと，出力が H になる頻度は，入力信号の大きさに比例するようになる．実際の出力は，図 6.30(b) のように，H や L の信号が続くことで結合し，PWM に類似の波形となる．

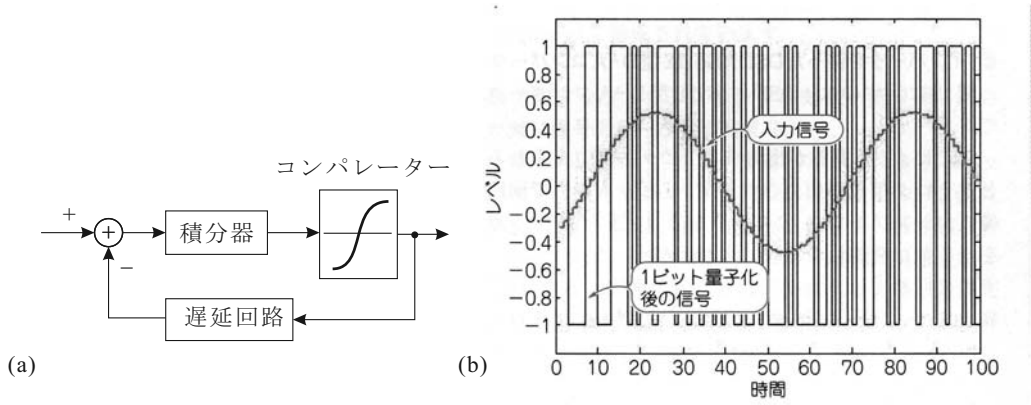


図 6.30 (a) Δ-Σ 型変調器の原理図．(b) アナログ入力（赤線) に対する Δ-Σ 変調信号の例．

6.6.10 適応フィルター

ディジタルフィルターの利点は，特性をダイナミックに変更できる，という点である．**適応フィルター** (adaptive filter) は，入力信号に応じて自動的に特性を変化させるものである．ごく平たくブロック図にしてしまうと図 6.31(a) のようになる．フィルタに斜めに矢印が付いているのは可変キャパシタや電圧源同様，フィルタの係数が変更可能，という意味である．

たとえば，FIR フィルタ (図 6.22) で各次の係数 a_i が外部信号で制御可能であれば，そこに制御信号を送るプログラムを書き，出力の様子を見ながらこれらを変更すれば良い．実際には IIR フィルタはこの用途に使用するうえで困難があり，専ら FIR フィルタが使用されている．

さて，図 6.31(a) では「所望信号」との差が小さくなるようにフィルター係数を調整するようになっているが，最初から「所望信号」がわかっていれば，わざわざフィルターなどを使用する必要ないはずである．実際の適応フィルターの用途の一例を図 6.31(b) に，これの変形版として示した．これは，**適応線スペクトル強調器** (adaptive line enhancer) と呼ばれる回路で，信号にランダムな雑音が重畳しているときに雑音除去のために使用する．不思議なことに，所望信号として雑音が混入した原信号そのものを使用している．しかし，フィルター通過信号の方には遅延回路が入っているため，信号に比して速い変化をする雑音は常に通過信号とは異なる成分として検出され，これを平均して小さくするようにフィルターを設計すれば，信号線が強調され，ノイズが除去されるフィルターが自然に形成される．

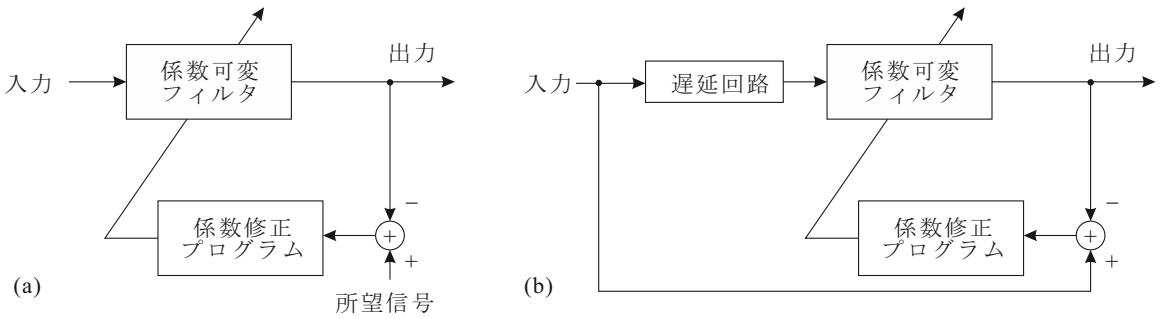


図 6.31　(a) 適応フィルターの概念図．(b) (a) を若干変形した線スペクトル強調器．

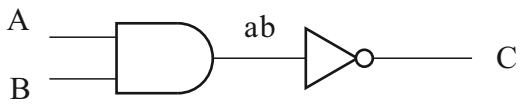
6.7　ハードウェア記述言語：HDL

最後に，時間もほとんどなくなってしまったが，ハードウェア記述言語 (hardware description language, HDL) について簡単に触れておこう．以上みたように，フィルターのような簡単なものでも多桁のシフトレジスタや掛け算器を大量に要し，これらをハードウェアで構成するのは容易ではない．一方，専用の LSI を設計するのは巨大なリソースを必要とする．そこで，PLD や FPGA(後述) を使い，プログラムによってこれらを個別に構成することが考えられてきた．

HDL は主にディジタル回路の動作仕様を記述するための (人工) 言語である．現在ではソフトウェア記述言語も並列処理を前提としたものが増えているが，HDL の場合は最初から経時的でかつ並列的な動作を記述できるように設計されている．その後 very high speed IC を冠して VHDL が作られ，またシミュレーションを強化する発展形として Verilog HDL が開発された．これらの言語は，ソフトウェアが高級言語→アセンブラ→機械語のような階層構造を持っているのに対応して高レベルから低レベルまでの階層構造を持つに至っている．

HDL 自身は，テキストで記述され，制御構造等ほぼ高級言語と同等に持っている一方，プログラム構造その他大きく異なっている．下の図は HDL で NAND ゲートを記述した例であるが，これでわかるように，プログラムはライブラリ宣言，エンティティ宣言，アーキテクチャ宣言の 3 部に分かれている．

```
-- Library declaration --------------
library IEEE;
use IEEE, STD_LOGIC_1164.ALL;
-- Entity declaration ---------------
entity NAND_CIRCUIT is
port(
A : in std_logic;
B : in std_logic;
C : out std_logc
);
end NAND_CIRCUIT;
-- Architecture declaration ----------
architecture RTL of NAND_CIRCUIT is
signal ab : std_logic;
begin
ab <= A and B;
C  <= not ab;
end RTL;
```



ライブラリ宣言

回路生成用のライブラリ，およびパッケージの宣言を行う部分で，use の後についているのがライブラリ中のパッケージの宣言である．

エンティティ宣言

回路中の NAND ゲートのような回路の「エンティティ」についての枠組みを示す．**generic** でプログラム中で使用する汎用値を宣言し，**port** によって外部回路との接続ポート，信号伝達方向を宣言する．

アーキテクチャ宣言

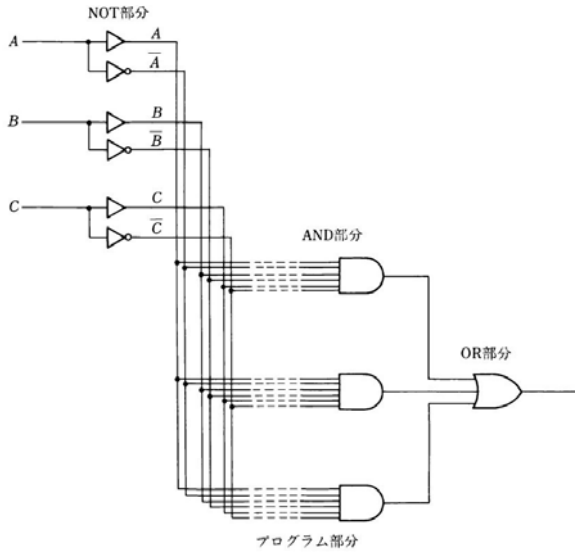
エンティティ宣言で示した各エンティティについて，ポート間でどのようなロジックによって動作するか，を規定する．“¡=” は信号の方向を決定する．また，begin - end で囲まれた内部は，論理関係のみが記述されていて処理の時間的な流れではない．原則として「同時進行」である．

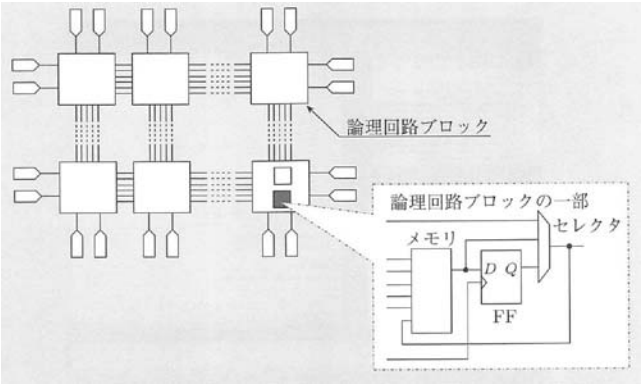
6.7.1 PLD，FPGA

HDL が成立するためのハードウェア上の基礎が programmable logic device (**PLD**) や，field programmable gate array (**FPGA**) と呼ばれる IC である．これらは，IC 内に多数の部品がアレイとして配置され，配線されており，ただし，配線の適当なクロス部分に「スイッチ」と称する結線・断線が可能なものが配置されている．並べ方は，図 6.32(a) のように PLD が組み合わせ論理回路を前提に NOT アレイと AND アレイを組み合わせ，これらを結線・断線する．これは，6.4.2 節で見たように，組み合わせ論理回路は一般に主加法標準展開で書けることに対応しており，回路リソースの許す範囲で一般の論理式を表現できることを意味する．

一方，FPGA の方は，非常に多種類の回路がこの名前で呼ばれているが，比較的多い構成として図 6.32(b) のように，ブロックに分かれたものをやはり多数の ON/OFF できる結線でつないでおり，ブロック内には FF が用意されていて順序回路も構成することができる．最近では，アナログ回路も混在するようなもの (mixed signal FPGA) も現れ，CPU をモノリシックに組み込みプログラムでロジックを表現できるようにするなど，多方面に発展しつつある．

配線を結線・断線するための方法としては，初期のころ，特に TTL を多く使用していた頃はヒューズ結線をして電流でヒューズを飛ばして断線することで行っていた．CMOS スイッチが一般化してからは，PROM，EPROM など 1 回書き込みのメモリが使われ，やがて EEPROM やフラッシュなどの書き換え可能メモリが使われるようになっている．





(a)

(b)

図 6.32　(a) PLD の回路構成の例．(b) FPGA の回路構成例

参考文献

[1] A. V. Oppenheim, R. W. Shafer, “Dicrete-Time Signal Processing” (Prentice Hall, 1999).

[2] M. H. Hayes, “Statistical Digital Signal Processing and Modeling” (Wiley and Sons, 1996).

[3] S. C. Douglas, “Introduction to Adaptive Filters” Ch.18 in *Digital Signal Processing Handbook* Eds. V. K. Madisetti and D. B. Williams (CRC Press, 1999).

[4] 小宮 浩 「高周波 PLL 回路のしくみと設計法」 (CQ 出版社，2011).