

# 物理屋のための電子回路論 第 13 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)

2016 年 1 月 17 日

## 7.5 A-D/D-A コンバータ

アナログ信号をデジタル信号に変換する、あるいはその逆の動作をする回路は、物理実験では極めて重要である。D-A/A-D 変換回路技術は、音響・映像技術にデジタル技術が入り込んで以来飛躍的に発展した。素子性能の向上による性能向上ばかりでなく、信号形式、回路方式の改革によって古い電子回路の教科書はみな書き換えなければならないほどの進歩が積み重ねられている。ただし、A-D/D-A 技術に関しては、CCD 技術、画素式ディスプレイの発達によりまず映像系からアナログ回路がほとんどなくなり、音響においても D 級アンプやスピーカーのデジタル駆動によりアナログ回路はデジタルのカプセルの向うの存在となりつつあることから、今後はこれまでのような急速な進展はないかもしれない。

「変換回路」と言っても、一方 (アナログ) が物理量であるのに対して他方 (デジタル) は、人間がこれを記述するために考えた記号であるから、「変換」のためには極めて様々なアイデアが存在しうる。これらは、変換形式に従い、外部クロックが必要になるなどの若干の使用法の違いはあるものの、A-D コンバータ (ADC)、D-A コンバータ (DAC) として LSI 化されてカプセル的に使用することが可能で、内部形式を知らなくても使えてしまう。それでも、どのようにして変換しているのか知っておくことは、電子回路の応用例としても悪くはない。代表的な回路形式を紹介しておこう。

### 7.5.1 デジタル-アナログ変換

まず、デジタルからアナログへの変換を考えよう。

#### 抵抗ストリング型、ラダー型

抵抗ネットワークを使用するものの内、図 7.14(a) に示した抵抗ストリング型は、余り現実性はないが、コンセプトとしては最も単純なものである。 $n$  ビットのデジタルデータは  $2^n$  の分解能を持っている。そこで、回路は標準電圧  $V_S$  を持っているとし、これを、 $2^n$  個の同一抵抗値  $R$  を持つ抵抗を直列につないだ回路に印可する (他端はグラウンド)。グラウンドから順番に抵抗に  $1, \dots, 2^n$  の番号を付けると、 $i$  番目と  $i+1$  番目の抵抗の間の配線部の電位は、 $i \times V_S/2^n$  となり、出力へは図のように、MOS スイッチを用いて接続する。受電部の入力インピーダンスを十分高くし、バイアス電流を抵抗ストリングに流す電流に比べて十分に低く取ることで  $i \times V_S/2^n$  をそのまま出力できる。

この方法は現実問題として、8 ビットでも 256 個の高精度抵抗と MOS スイッチを結線する必要があり、デコーダーも同じだけの数の出力を要するなど、使いにくいものである。

これに対して実際にもよく使用されるのが図 7.14(b) の抵抗ラダー型 DA 変換である。 $n-1$  個の抵抗値  $R$  の抵抗と  $n+2$  個の値  $2R$  の抵抗を用意し、 $n$  ビットの入力ラインを図のように相反性 MOS スイッチのゲートに接続し、 $2R$  の抵抗を介して  $R$  の抵抗を直列にした各端点に接続する。最低ビットラインの端点は  $2R$  で接地し、他端はやはり  $2R$  でこちらは OP アンプの反転増幅器の  $-$  入力への入力抵抗として使用する。

OP アンプの項で学んだように、 $-$  入力端子は仮想接地されているので、ここでは右端の  $2R$  もやはり接地されて

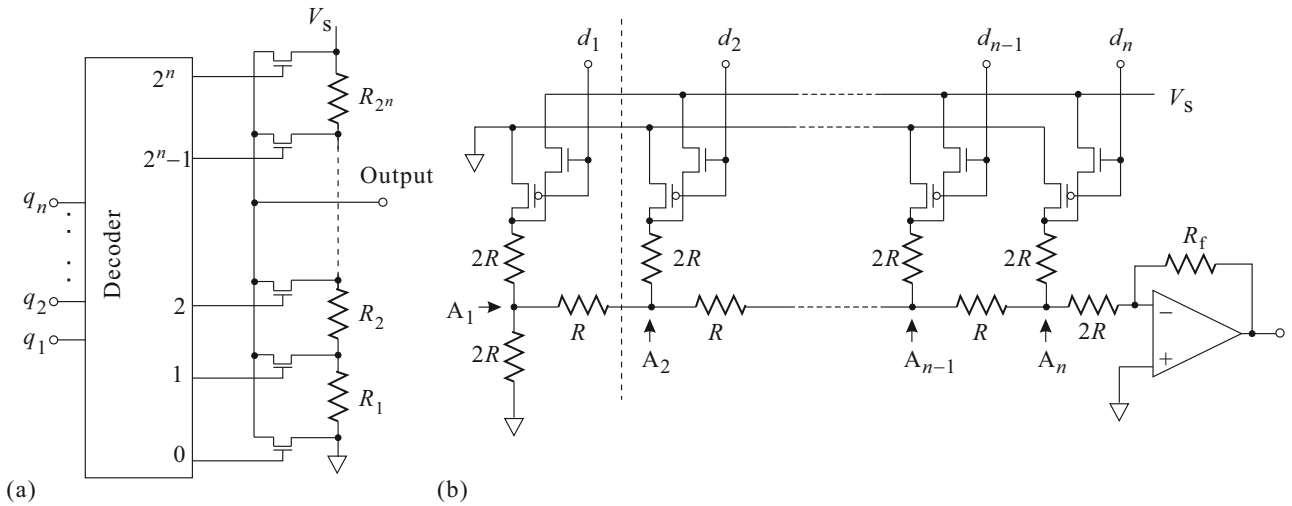


図 7.14 抵抗ネットワークを用いた D-A コンバータの等価回路図. (a) 抵抗ストリング型. (b) 抵抗ラダー型. FET スイッチは, ゲートに○がついていないものが n チャネルで H 入力に対して ON, 反転記号である○が付いているものが p チャネルで H 入力に対して OFF となる.

いるものとして  $A_n$  点の電位を考える. 今,  $k$  ビット目が 1(H) で他はすべて 0(L) が入力されたとする.  $A_n$  点から右および上方の接地側を見ると  $2R$  の抵抗が 2 個並列になっており, 合成抵抗  $R$  で接地されている. 従ってその左の  $A_{n-1}$  点から右方向の  $A_n$  点を通る経路の接地までの合成抵抗はやはり  $2R$  で  $A_n$  点と状況が全く同じであることがわかる. これが,  $A_{k+1}$  点まで続く.  $A_{k-1}$  点より左側を見ても, 全く同じ状況である. 従って  $A_k$  点は, 左右に  $2R$  で接地され,  $V_S$  より  $2R$  を通して接続されていることがわかる. 結局流れる全電流は  $V_S/3R$ , 片方の分岐に流れるのはその  $1/2$  である. 以下, 分岐ごとに  $1/2$  になるので, 最終的に OP アンプ方向へ流れ出す電流は

$$J_{\text{out}} \left( \begin{matrix} 0 \cdots 0 & 1 & 0 \cdots 0 \\ n & k & 1 \end{matrix} \right) = \frac{V_S}{3R} \left( \frac{1}{2} \right)^{n-k+1} = \frac{V_S}{6 \cdot 2^n R} 2^k$$

となる.  $J_{\text{out}}$  の引数は, 入力を 2 進数で表したものになっている. 従って, 入力  $\{d_i\}$  ( $d_i = 0$  or  $1, i = 1, \dots, n$ ) があつた時の OP アンプ出力は, 重ね合わせの定理 (2.5 節) を用いて,

$$V_{\text{out}}(\{d_i\}) = -\frac{1}{3 \cdot 2^n} \frac{R_f}{2R} V_S \sum_{k=1}^n 2^k d_k \quad (7.11)$$

となつて, AD 変換がされている事がわかる. ストリング型に比べて必要抵抗は圧倒的に少なく, デコーダなども必要としない. 実際多用される形式である.

この他, 受動素子ネットワークを用いるものとしては, 容量アレイ型などがある.

#### 電流加算型

原理的にはたいへん簡単な変換法であり, レポート問題として考えてもらう予定.

#### 容量アレイ型

これも簡単なので, レポート問題として考えてもらう予定である.

#### パルス幅変調型

パルス幅変調 (pulse width modulation, PWM) は, どちらかと言うとアナログ信号を PWM 信号にしてパワートランジスタをスイッチすることで電力増幅し, ローパスフィルターでアナログ信号に戻す, アナログ電力増幅に用いることが多い. が, 無論,

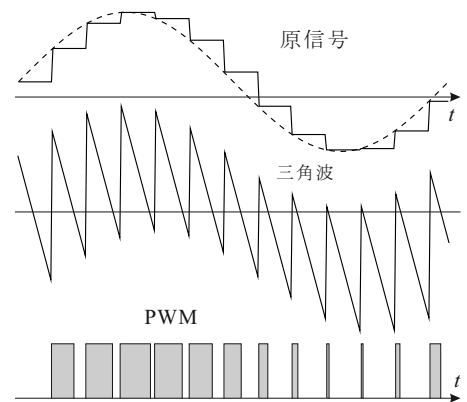


図 7.15 パルス幅変調法の一例.

速くて正確なクロック信号とカウンタを用いてデジタル的に PWM 信号を作り出すことができ、これをローパスフィルタに通すことで DA 変換ができる。

原理は以上で終了、という位簡単明瞭である。逆に PWM 信号を作り出す方法の一例を図 7.15 に示した。クロックに同期したのこぎり三角波を作り出し、原信号でベースラインをシフトさせる。これをコンパレータに入れてこの変調三角波が閾値電圧を超えた領域だけ H 信号を出すようにすることで、PWM 信号が得られる。

## 7.5.2 アナログ-デジタル変換

### 逐次比較型

この方式では標本回路 (sample and hold circuit) により入力電圧を固定しておき (これは、他の多くの方式でも用いられる)、DA 変換によって既知電圧を作り出し、これとホールドした電圧とが一致するかどうかを調べることでデジタルデータに変換する。様々な意味での「既知電圧」を次々とホールド電圧と比較することから**逐次比較** (successive approximation) 型と呼ばれている。

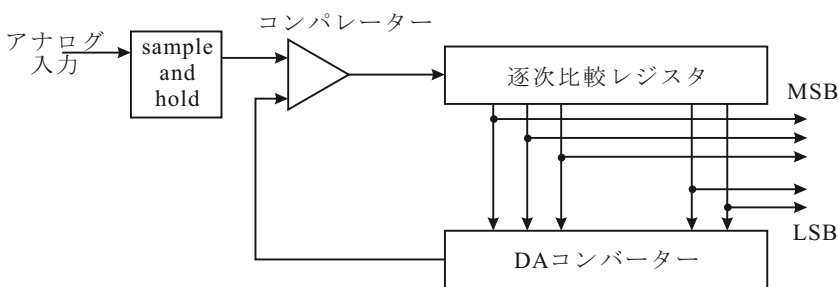
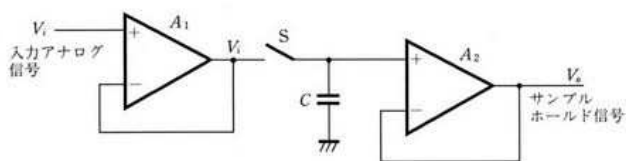


図 7.16 逐次比較型 AD コンバータのブロックダイアグラム

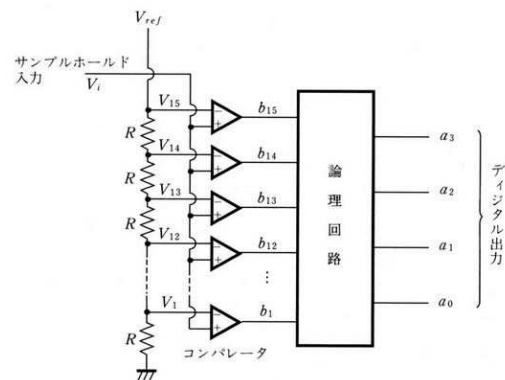


サンプルアンドホールド回路 (AD 変換回路ではトラックアンドホールドと呼ばれることもある) にも多くの形式があるが、左図に挙げるものが最も簡単である。信号によってキャパシタにチャージし、これを OP アンプにより孤立状態にして維持する。新しくサンプルする際には、放電するのが一般的であるが、信号

時間変化が緩やかでかつ低出力抵抗である場合は、帯電状態で接続することも考えられる。

2 進数に変換することから、逐次比較は高いビットから、数値計算でいう 2 分法によって行うのが一般的である。最初に最高ビットのみを立てた参照電圧を DAC により用意し、コンパレータで比較する。コンパレータもまた、OP アンプを開ループ (可能な限り大きな開ループゲインを持たせる) で使用するのが精度的に有利である。参照電圧がサンプルより大きければ 0、小さければ 1 が立つ。このデジタル信号の次のビットを立てたものを DAC に入れて次の参照電圧を作る。以下最終ビットまで繰り返すことでデジタル変換が完了する。

### フラッシュ型



逐次比較型は 1 ビットごとに変換していくので変換回路は 1 回路で済むが、多ビットの AD 変換を行う場合にはビット分だけのサイクルを繰り返す必要があり時間がかかるようになる。これに対して、多数のコンパレータを並べて一気に AD 変換を行おうというのがフラッシュ型 ADC である。左図のように、抵抗ストリング型 DAC のちょうど逆のような形式である。

### パイプライン型

フラッシュ型は逐次比較型に比べて圧倒的な速度を持っているが、当然ながら分解能分の分割抵抗とコンパレータという消費リソースも巨大となる。また、精度限界はコンパレータの開ループゲイン  $A$  で決まることが多い。例えば、 $A$  が 100dB だと

すると、入力  $10\mu\text{V}$  で出力は  $1\text{V}$  になり、デジタル回路が TTL レベルで動作しているとする、しきい値電圧  $V_{th}$  には到達しない。さりとて、アナログアンプで例えば 10 倍に増幅して入力しようとする、アンプの出力、あるいはコンパレーターの入力の飽和値によってアナログ電圧の大きさが制限されてしまう。

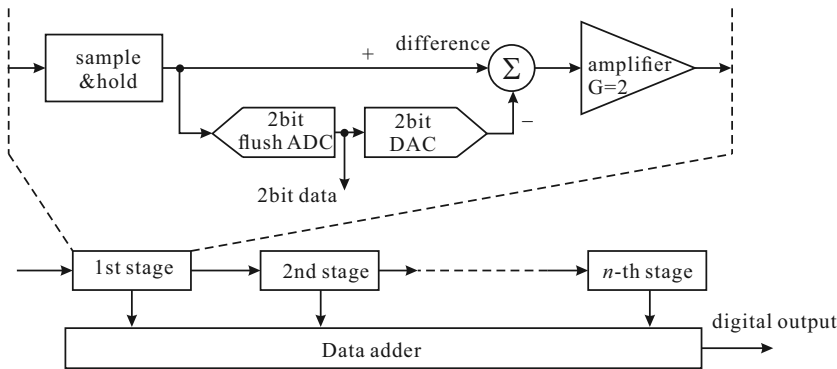


図 7.17 2 ビット単位パイプライン型 ADC のブロック図

このような欠点を、速度を若干犠牲にしながらカバーする方式が幾つか使われており、パイプライン型はその一つである。図 7.17 はそのブロック図 (後述) で、数ビットをまとめてフラッシュ型で AD 変換し、残った信号を増幅器で増幅しながら次のサイクルを行うようになっている。フラッシュ型と逐次比較型の間の子のような形式である。

### 積分型

逆に時間はかかるができるだけ少ない部品リソースで AD 変換を行おうというのが積分型 ADC で、積分回路に定電流を流して時間に対してリニアに増加する電圧を作り出し、入力電圧とコンパレーターで比較する。増加電圧が下端からスタートする際にデジタルカウンタをスタートし、増加電圧が入力電圧をクロスしてコンパレーターが ON になった時点でストップする。カウンタ出力により AD 変換ができる。積分回路とコンパレーターの OP アンプと積分用のキャパシタで精度がほぼ決まり (カウンタの桁数を出すのは比較的容易)、安価に AD 変換を構成できるため、パネルメーター (機器の表面に沢山はめ込んで、様々なパラメーターをモニターできるようにするもの) などに良く使用されている。

### デルタシグマ型

オーバーサンプリング型と呼ばれる ADC の代表的な形式である。1 ビット (従ってコンパレーター 1 個のみ) の超低分解能 ACD を用いフィードバックを使ったデルタシグマ変換により高精度デジタル信号に変換する。詳細は後述。

## 7.5.3 情報の符号化

「デジタル信号序論」で述べたように、パルス振幅変調 PAM でパルス高を離散化すれば、デジタル信号になる。それをそのまま変調の形にしたのがパルス符号変調 (pulse code modulation, PCM) である。PCM データは、クロックに合わせて何ビットかのデータを伝送するもので、少ない伝送チャネルで送る場合は、時系列信号にしてシリアル伝送する。復調する必要がある場合は、パラレルデータに変えて DAC によりアナログ信号にする。

この直接的な PCM 法に対して、様々な変形がある。デジタル信号は符号化されたものであるため、急激な信号値変化もデータ上は表現できるが、アナログ信号は、特にそれに含まれている周波数成分の上限で表現できるよりも大きな変化は生じない。そこで、値そのものではなく前の信号値との差分を抽出して伝送することも考えられる。これを差分 PCM (differential PCM, DPCM) 方式と呼び、予測符号化方式の一種である。DPCM は一般に PCM より小さなビットレートで同精度の信号を伝送することができる。「予測」という文字がついているのは、実際には単なる差分ではなく、最も単純な前値予測方式であっても予測係数がついて、受信データ  $\{x_i\}$  に対する予測値は  $x'_i = ax_{i-1}$  で与えられ、送付データは差分  $\epsilon_i = x_i - ax'_i$  である。

また、PCM でない符号化形式 (時間領域で変換してから符号化するもの) など、符号化にも沢山の形式があるが、とても踏み込む余地がないので本講義ではここまでとする。

## 7.6 デジタルフィルター

一旦デ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、デジタル信号からデジタル信号への変換処理を**フィルター (filter)** 作用と呼ぶ。デジタル信号処理 (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「デジタル信号処理」という意味、および「デジタル信号プロセッサ」(digital signal processor)の両方に用いられ、混用されている。

デ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、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。

### 7.6.1 ブロック・ダイアグラム

デ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  $\{x_i\} = (x_0, x_1, \dots)$  に対して、別の数列  $\{y_i\} = (y_0, y_1, \dots)$  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

$$y_n = F(x_{n-k}, x_{n-k+1}, \dots, x_n) \quad (7.12)$$

と書けることになる。

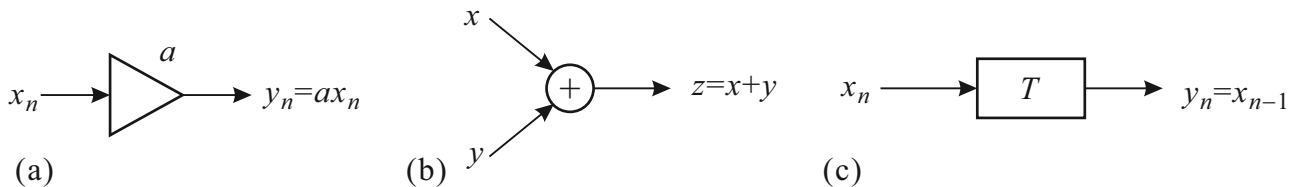
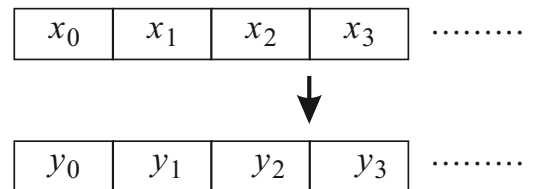


図 7.18 デジタルフィルター用ブロックダイアグラムに使われる、「基本ブロック」。(a) 定数倍回路。(b) 加算回路。(c) 遅延回路 (シフトレジスタ)。

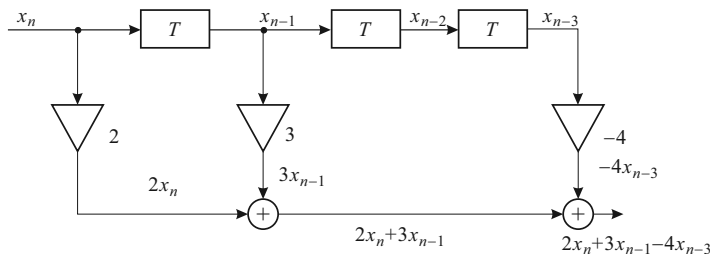


図 7.19 式 (7.13) のダイアグラム化。  
 タル回路のバッファの形をしているが、記入されている数倍だけする、**定数倍回路**、である。(b) は信号の和を取る**加算回路**である。(c) は、信号を時間領域で1単位だけ遅らせる**遅延回路**である。

$F$  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 7.18 の3つであり、(a) はディ

例として

$$y_n = 2x_n + 3x_{n-1} - 4x_{n-3} \quad (7.13)$$

をダイアグラム化すると前頁の図のようになる。

### 7.6.2 z 変換空間表示

デ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、デジタルフィルタも (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 (7.13) を考える。今の場合、フィルタの入出力の z 変換は定義 (6.83) より、

$$X(z) = \sum_{n=0}^{\infty} x_n z^{-n}, \quad Y(z) = \sum_{n=0}^{\infty} y_n z^{-n} \quad (7.14)$$

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  $\tau$  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

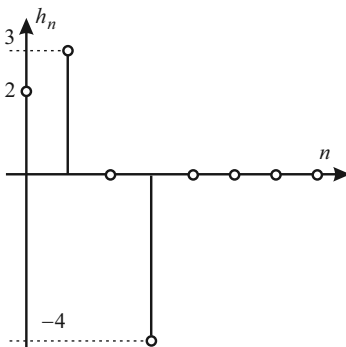
$$Y(z) = 2X(z) + 3z^{-1}X(z) - 4z^{-3}X(z) = (2 + 3z^{-1} - 4z^{-3})X(z) \quad (7.15)$$

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、(6.89) より

$$H(z) = 2 + 3z^{-1} - 4z^{-3} \quad (7.16)$$

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、このデジタルフィルタ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。一般にフィルタの関数  $F(7.12)$  が、



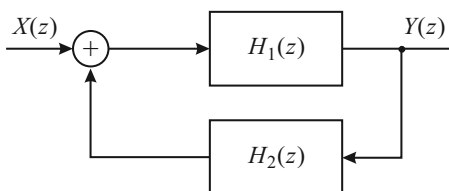
$$F(x_{n-k}, x_{n-k+1}, \dots, x_n) = \sum_{j=0}^k a_j x_{n-j} \quad (7.17)$$

のようにかける場合、伝達関数は

$$H(z) = \sum_{j=0}^k a_j z^{-j} \quad (7.18)$$

となる。

### 7.6.3 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

$$H(z) = \frac{H_1(z)}{1 - H_1(z)H_2(z)} \quad (7.19)$$

これは、むしろ、アナログ回路の場合の (4.3) 式に対応するもので、フィードバック回路の場合、一般に

$$(\text{全体の伝達関数}) = \frac{(\text{直接部分の伝達ゲイン})}{1 - (\text{フィードバック部分の伝達ゲイン})} \quad (7.20)$$

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 (7.12) 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。

### 7.6.4 デジタルフィルターの分類

デジタルフィルターは、上のようにフィードバックを持つために (7.12) 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、バターワース、チェビシェフ、などの分類が可能である。

### 7.6.5 FIR フィルター

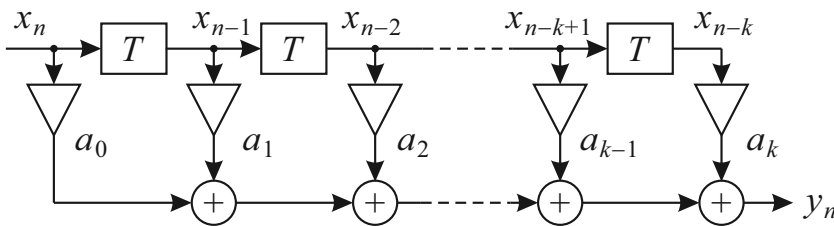


図 7.20 FIR フィルターの一般形 (7.17) のブロックダイアグラム。

線形 FIR フィルターは、一般に (7.17) の関数で表され、図 7.20 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 (7.18) には極がなく安定である。FIR フィルターの周波数特性は、(7.18) より、 $z = e^{i\omega\tau}$  とおいて、

$$H(e^{i\omega\tau}) = \sum_{j=0}^k a_j e^{-ij\omega\tau} \quad (7.21)$$

である。

まず、もっとも単純な移動平均、および隣接差分、

$$F_{\pm}(x_n, x_{n-1}) = (x_n \pm x_{n-1})/2 \quad (7.22)$$

については、

$$H_{\pm}(e^{i\omega\tau}) = e^{-i\omega\tau/2} \begin{pmatrix} \cos(\omega\tau/2) \\ i \sin(\omega\tau/2) \end{pmatrix} \quad (7.23)$$

となる。

$\omega$  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  $\omega$  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

$$F_d = [(x_n + x_{n-1}) - (x_{n-1} + x_{n-2})]/2 = [x_n - x_{n-2}]/2 \quad (7.24)$$

は、

$$H_d = (1 - e^{-2i\omega\tau})/2 = ie^{-i\omega\tau} \sin \omega\tau \quad (7.25)$$

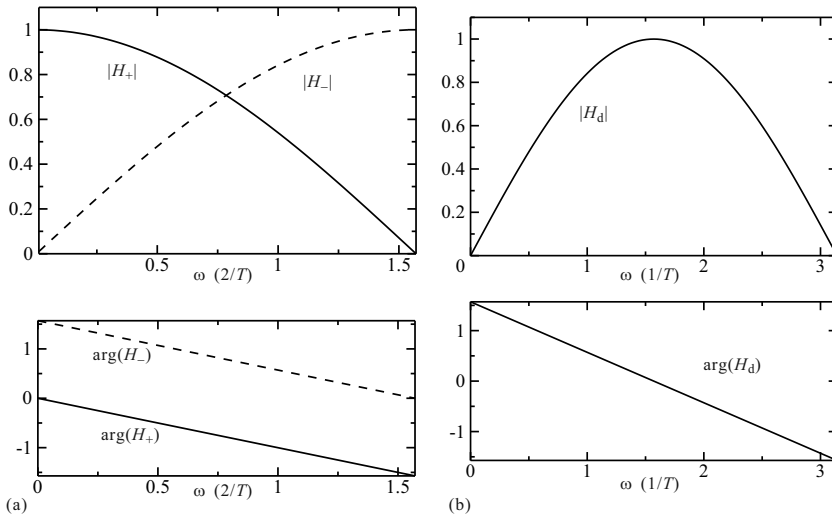


図 7.21 (a) 移動平均, 隣接差分操作によるフィルターの周波数応答 (上が伝達関数の絶対値, 下が位相) 特性.  $\omega$  は  $2/\tau$  単位で計っているの  
ので,  $\pi/2$  以下の領域に限られる. (b) 移動平均の差分に対して (a) と同様の周波数特性. ただし,  $\omega$  は  $1/\tau$  目盛り.

より, 図 7.21(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.

### 7.6.6 IIR フィルター

本節冒頭で「デ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,

$$\lim_{n \rightarrow \infty} y_n = 0, \quad (7.26)$$

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.

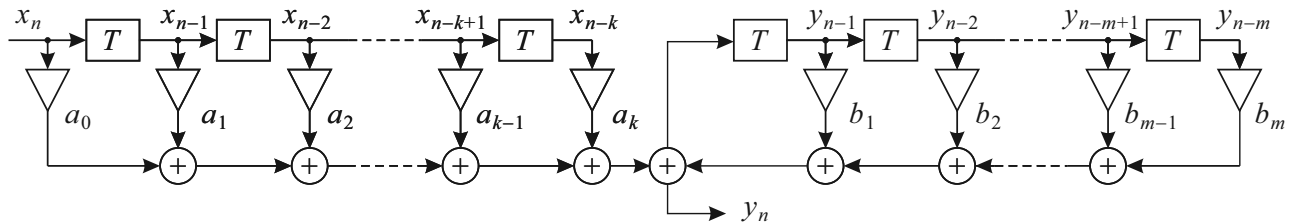


図 7.22 IIR フィルターの一般形ダイアグラム

IIR フィルタをダイアグラムで一般的に表すと, 図 7.22 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

$$y_n = \sum_{l=0}^k a_l x_{n-l} + \sum_{j=1}^m b_j y_{n-j} \quad (7.27)$$

のように書ける. この両辺を  $z$  変換すると,

$$Y(z) = X(z) \sum_{l=0}^k a_l z^{-l} + Y(z) \sum_{j=1}^m b_j z^{-j} \quad (7.28)$$

であるから, 伝達関数は

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{l=0}^k a_l z^{-l}}{1 - \sum_{j=1}^m b_j z^{-j}} \quad (7.29)$$



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 (7.26) 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、(7.29) 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。

### 7.6.7 デジタルフィルタの設計

デジタルフィルタの設計には様々な方法が存在する。FIR フィルタの場合、Remez 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 remez 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  $\omega_c$  以下は完全透過し、以上を完全カットするもので、

$$G(e^{i\omega\tau}) = \begin{cases} 1, & |\omega| \leq \omega_c, \\ 0, & \omega_c < |\omega| \leq \omega_N \end{cases} \quad (7.30)$$

と表される。 $\omega_N = \omega_s/2 = \pi/\tau$  はナイキスト周波数である。(7.30) は 5.3 節で見たように、 $[-\omega_N, \omega_N]$  の範囲で考えることにすればこの外側では  $2\omega_N = \omega_s$  の周期を持つ周期関数としてフーリエ級数展開し、

$$G(e^{i\omega\tau}) = \frac{\omega_c}{\omega_N} \sum_{n=-\infty}^{\infty} \frac{1}{n\pi} \text{sinc}\left(n \frac{\omega_c}{\omega_N}\right) e^{-ni\omega\tau} = \gamma_c \sum_{n=-\infty}^{\infty} \frac{1}{n\pi} \text{sinc}(n\gamma_c) z^{-n} \quad (7.31)$$

が得られる。 $\gamma_c \equiv \omega_c/\omega_N$  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  $\gamma$  として sinc 関数  $\text{sinc}(n\gamma_c)$  で与えられることがわかる。

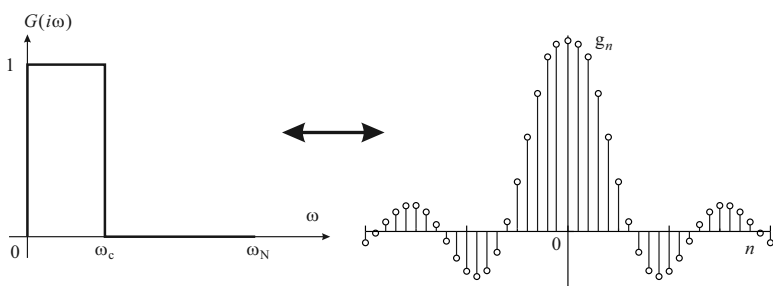


図 7.23 理想低域通過フィルタ (LPF) の周波数特性模式図 (左) と、これをフーリエ級数展開した際の  $n$  次項  $\text{sinc}(n\omega_c/\omega_N)$  の模式図 (右)。

FIR フィルタで理想 LPF を実現しようとする、(7.31) のように無限次数のフィルタが必要となる。sinc 関数  $\text{sinc}(x)$  は、 $x \rightarrow \infty$  に対して  $1/x$  で小さくなるので有限次数で打ち切ると、図 7.24(a) のように周波数特

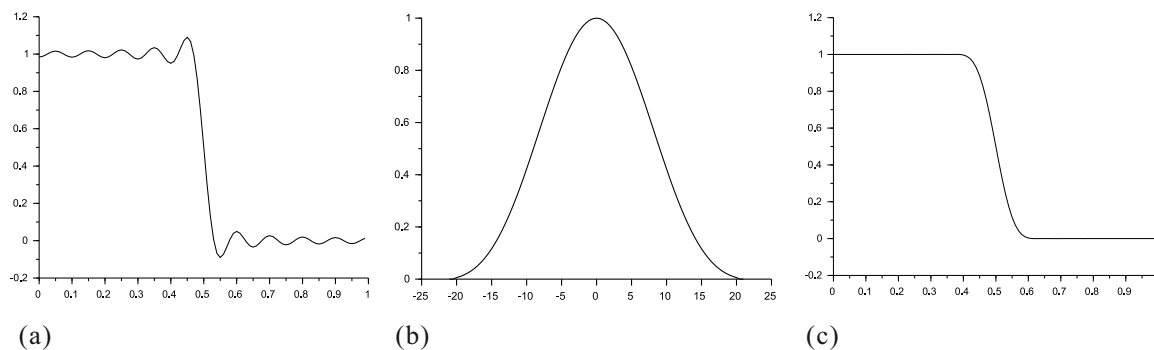


図 7.24 (a)(7.31) の理想 LPF の次数を  $n = 20$  で打ち切ったフィルタの周波数特性。 $\omega/\omega_N$  を横軸にとった。また、 $\omega_c/\omega_N = 0.5$  とした。(b) カイザー窓関数の例。 $\alpha = 7$ ,  $L = 20$  として (7.32) をプロットしたもの。(c) (a) のフィルタに、(b) の窓関数を乗じたフィルタの周波数特性。

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

$$w_n = \begin{cases} \frac{I_0(\alpha\sqrt{1-(n/L)^2})}{I_0(\alpha)} & |n| \leq L, \\ 0 & |n| > L \end{cases} \quad (7.32)$$

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。 $\alpha$ ,  $L$  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 7.24(b) にその形状例を示した。これを (7.31) に乗じると、図 7.24(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ーワース (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、伝達関数がある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。バターワース特性を持つ  $n$  次の LPF の伝達関数は

$$\Xi(s) = \sum_{k=0}^{N-1} \frac{\omega_k}{s - s_k}, \quad s_k = r_c \exp \left[ i \left\{ \frac{\pi}{2} + \frac{(2k+1)\pi}{2n} \right\} \right] \quad (7.33)$$

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  $u_H(t)$  と書いて

$$\xi(t) = u_H(t) \sum_{k=0}^{n-1} w_k \exp(s_k t) \quad (7.34)$$

である。これを  $\tau = 1$  で標準化 (離散化) すると、

$$h_n = h_{Hn} \sum_{k=0}^{n-1} w_k e^{ns_k}, \quad \therefore H(z) = \sum_{k=0}^{n-1} \frac{w_k}{1 - \exp(s_k)z^{-1}}. \quad (7.35)$$

これは、(7.33) で

$$(s - s_k)^{-1} \rightarrow (1 - \exp(s_k)z^{-1})^{-1} \quad (7.36)$$

という置換を施した形である。(7.36) 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。

これに対して双一次  $z$  変換法 (bilinear  $z$ -transform) では、

$$s \rightarrow \frac{1 - z^{-1}}{1 + z^{-1}} \quad (7.37)$$

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。

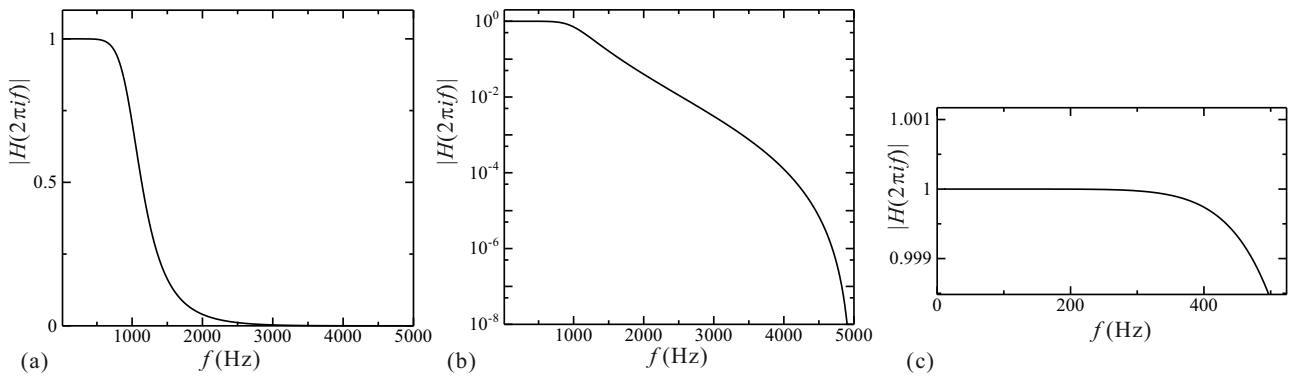
図 7.25 はこのようにして、4 次のバターワース特性フィルタについて設計したもので伝達関数は

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3} - a_4 z^{-4}} \quad (7.38)$$

のように書け、係数は図 7.25 中のテーブルのようになる。図にも示したように、高々 4 次のフィルタでリップルが全くない特性を作ることができる。

## 7.6.8 複素信号処理と位相ロッキング

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、デジタルの場合は同期回路であれば多次元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  $\omega$  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



coef.	0	1	2	3	4
$a_n$		2.3695130072	-2.3139884144	1.0546654059	$-1.8737949237 \times 10^{-1}$
$b_n$	$4.8243433577 \times 10^{-3}$	$1.9297373431 \times 10^{-2}$	$2.8946060146 \times 10^{-2}$	$1.9297373431 \times 10^{-2}$	$4.8243433577 \times 10^{-3}$

図 7.25 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, (7.38) 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.

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 7.26(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 7.26(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.

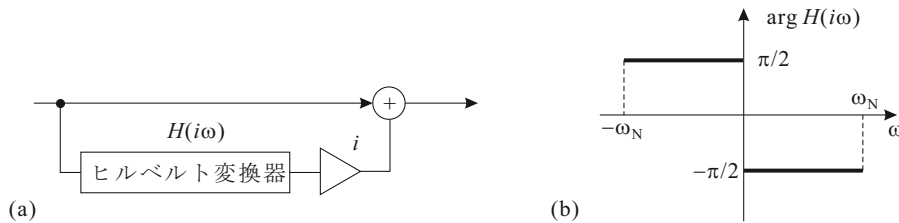


図 7.26 (a) ヒルベルト変換器による複素信号生成回路のブロックダイアグラム. (b) ヒルベルト変換器位相の  $\omega$  依存性.

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 7.27 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.

ここではデ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 7.27(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  $\omega_0$  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  $\omega_0$  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

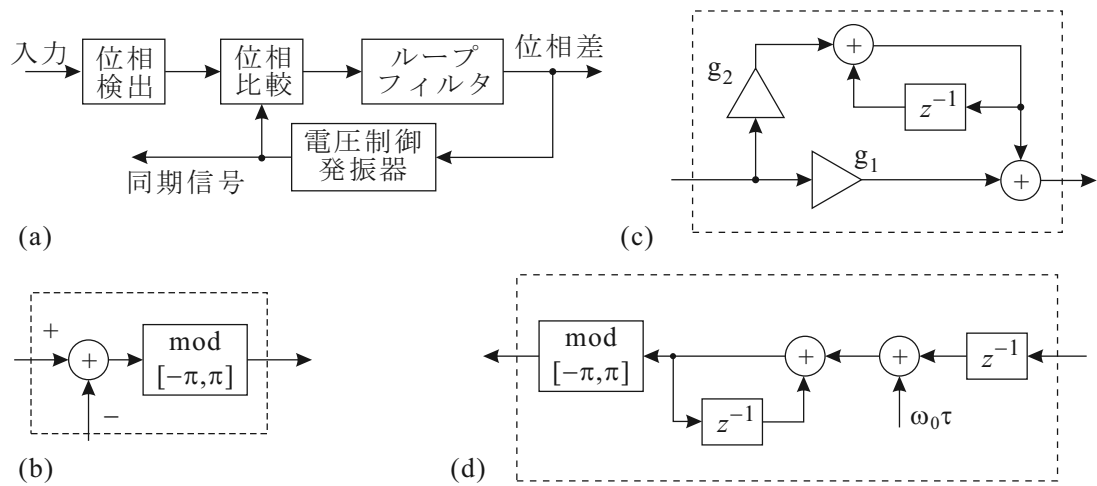


図 7.27 (a) 位相ロック回路のブロックダイアグラム. (b) ループフィルタ部分. (c) 位相比較部分. (d) 電圧制御発振器部分.

まず、通常の実信号入力を図 7.26(a) の回路を通すことで複素信号化 ( $x_n^{(r)}, x_n^{(i)}$ ) し、位相データ

$$\phi_n = \arctan \frac{x_n^{(i)}}{x_n^{(r)}} \quad (7.39)$$

を得る. 同様のことを VCO の出力に対して行い、位相データ  $\varphi_n$  を得、図 7.27(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  $2\pi$  の加減算を行う.

#### ループ・フィルタ

位相比較出力が送られるループ・フィルタは図 7.27(c) のような IIR フィルターで、伝達関数は

$$H(z) = g_1 + \frac{g_2}{1 - z^{-1}} \quad (7.40)$$

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  $\omega_v$  は入力信号周波数  $\omega_i$  にロックされ、 $\omega_v = \omega_i$  となる. (7.26) が満たされていないため、ループ・フィルタは必ずしも安定ではない. 安定のためには条件

$$g_2 < -2g_1 + 4 \quad (7.41)$$

が満たされる必要がある.

#### VCO

図 7.27(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、周波数通倍、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.

### 7.6.9 $\Delta$ - $\Sigma$ 型 A-D コンバーター

$\Delta$ - $\Sigma$  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ジタル信号を作り出してしまふ. 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 7.28(a) には、ごく簡単な原理を示してある. コンパレータはあくまで 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 7.28(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。

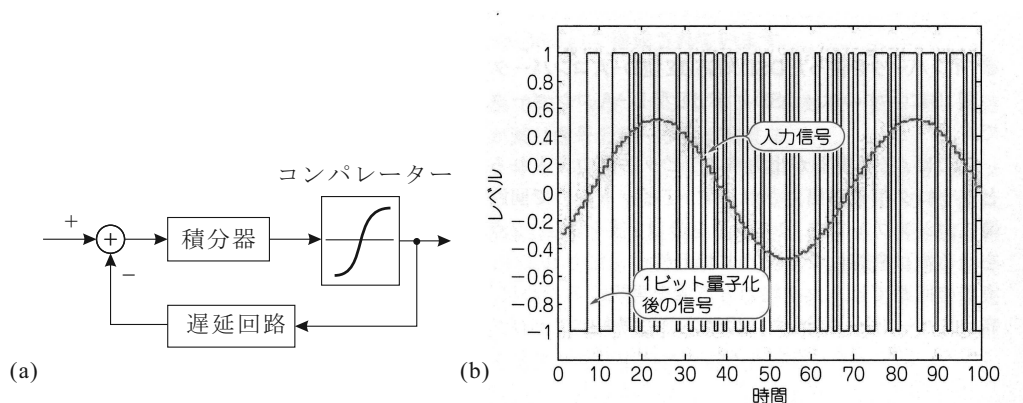


図 7.28 (a)  $\Delta$ - $\Sigma$  型変調器の原理図. (b) アナログ入力 (赤線) に対する  $\Delta$ - $\Sigma$  変調信号の例.

### 7.6.10 適応フィルター

デ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 7.29(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 7.20) 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 7.29(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」がわかっていたら、わざわざフィルターなどを使用する必要はないはずである。実際の適応フィルターの用途の一例を図 7.29(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。

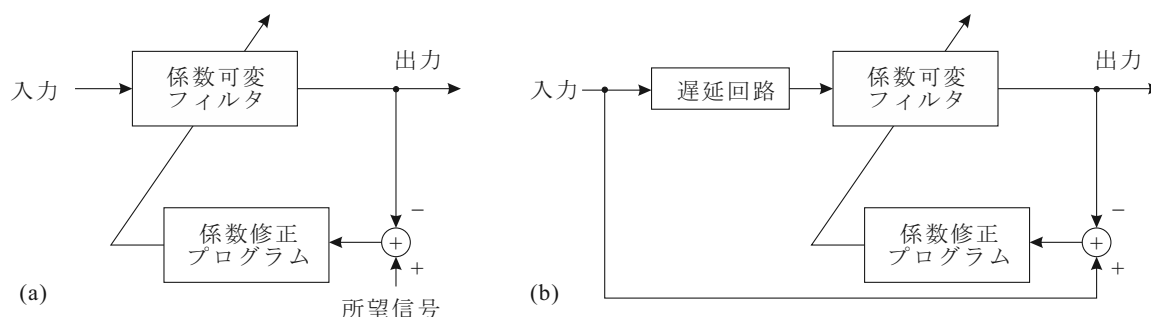


図 7.29 (a) 適応フィルターの概念図. (b) (a) を若干変形した線スペクトル強調器.

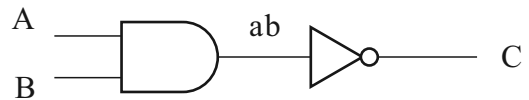
## 7.7 ハードウェア記述言語 : HDL

最後に、時間もほとんどなくなってしまうが、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きているが、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部分に分かれている。

```
-- 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_logic  
);  
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。

## 7.7.1 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。(FPGA は広義の PLD に含まれる。) 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 7.30(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は呼ばれているが、比較的多い構成として図 7.30(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く。

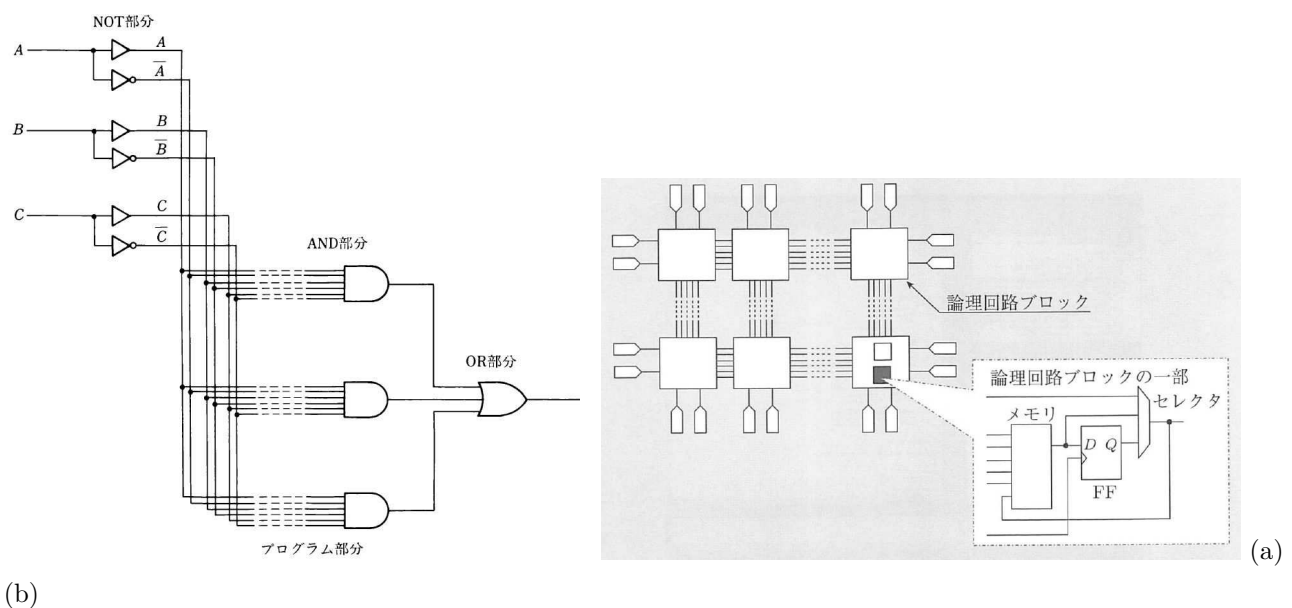


図 7.30 (a) PLD の回路構成の例. (b) FPGA の回路構成例

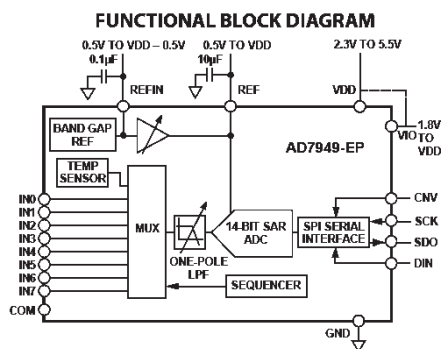
## 参考文献

- [1] A. V. Oppenheim, R. W. Shafer, “Discrete-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).

## 付録 K : A-D/D-A コンバータ選択・使用上の注意

現在の物理実験家が自ら A-D/D-A 回路を設計製作することはまず考えられないが、集積回路の形で用いる可能性はある。デジタルオーディオの発展に伴って驚異的な多ビットの DAC などが二束三文で売られるようになり、これらを上手に使用することで高速/高精度の実験測定系が安価・低労力で得られる可能性がある一方、絶対精度等には注意が必要である。

現在オーディオ用に 24 ビットの DA 変換 IC は極めて安価に手に入る。AD 変換も同様であり、中には 32 ビットのような高いビット数を持つものもある。これらはほとんど  $\Delta$ - $\Sigma$  型の AD 変換器を用いており、相対精度が非常に高く便利である。が、絶対精度はあまり当てにできないので、計測に用いる場合は精密機器による較正が必要である。



実際に ADC/DAC を選択する場合、データシートをきちんと見る必要がある。まず、どのような回路が使用されているのか把握し、トラブルが生じた際に備える (回路形式が原因の可能性ある)。これには、ブロック図を見ておけば十分で、データシートには大抵簡単なブロック図が掲載してある。左図は AD7949 の場合であり、中に MUX と書かれた 8 チャンネルのマルチプレクサ (入力チャンネル切替器) が入っており、ボード線図で「極」が 1 つだけある低域通過フィルター (LPF) が入れられていることがわかる。その先は「14bit SAR ADC」につながっていて、この IC が 14 ビットの抵抗逐次比較型 (successive approximation resistor) の ADC であることがわかる。

次に見なければならぬのが specification table で、面倒でも定格電圧やノイズ、不確定性の最悪値などをチェックし、実験の用途に耐えるものかどうかを調べておく必要がある。



## SPECIFICATIONS

VDD = 2.3 V to 5.5 V, VIO = 1.8 V to VDD, V<sub>REF</sub> = VDD, all specifications -55 °C to +125 °C, unless otherwise noted.

Table 2.

Parameter	Conditions/Comments	Min	Typ	Max	Unit
RESOLUTION		14			Bits
ANALOG INPUT					
Voltage Range	Unipolar mode	0		+V <sub>REF</sub>	V
	Bipolar mode	-V <sub>REF</sub> /2		+V <sub>REF</sub> /2	V
Absolute Input Voltage	Positive input, unipolar and bipolar modes	-0.1		V <sub>REF</sub> + 0.1	V
	Negative or COM input, unipolar mode	-0.1		+0.1	V
	Negative or COM input, bipolar mode	V <sub>REF</sub> /2 - 0.1	V <sub>REF</sub> /2	V <sub>REF</sub> /2 + 0.1	V
Analog Input CMRR	f <sub>IN</sub> = 250 kHz		68		dB
Leakage Current at 25°C	Acquisition phase		1		nA
Input Impedance <sup>1</sup>					
THROUGHPUT					
Conversion Rate					
Full Bandwidth <sup>2</sup>	VDD = 4.5 V to 5.5 V	0		250	kSPS
	VDD = 2.3 V to 4.5 V	0		200	kSPS
¼ Bandwidth <sup>2</sup>	VDD = 4.5 V to 5.5 V	0		62.5	kSPS
	VDD = 2.3 V to 4.5 V	0		50	kSPS
Transient Response	Full-scale step, full bandwidth			1.8	µs
	Full-scale step, ¼ bandwidth			14.5	µs
ACCURACY					
No Missing Codes		14			Bits
Integral Linearity Error		-1	±0.5	+1	LSB <sup>3</sup>
Differential Linearity Error		-1	±0.25	+1	LSB

図 7.31 AD7949 の specification table の一部

最後に、A-D 変換方式と分解能、変換速度の「分布図」をあげておく。

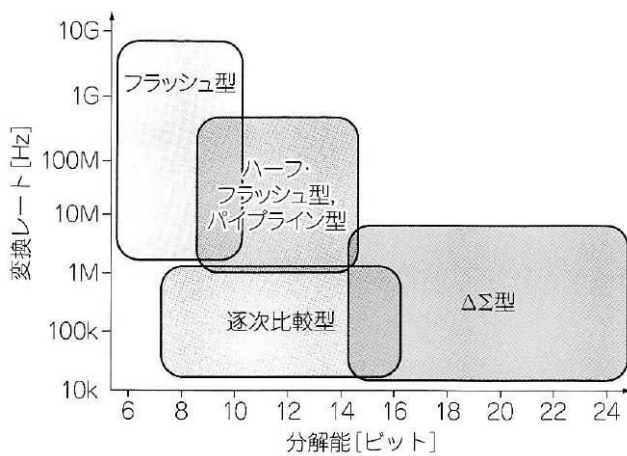


図 7.32 A-D 変換器の回路方式による、分解能、変換速度に対する分布図。