第6章 設計に広がりをもたせる

これまで，ディジタル信号処理の基本となるディジタル・フィルタ処理の説明をしました．ディジタル信号処理ではディジタル・フィルタは頻繁に登場しますが，それがすべてではありません．そのほかよく使われる処理に高速フーリエ変換を含むさまざまな変換処理があります．フーリエ変換の基礎的な部分はすでに説明しました．

ここでは，それ以外のさまざまな信号処理として使う変換について説明します．

6-1 高速フーリエ変換（FFT）

6-1-1 周波数解析を可能とする強力なツール

ディジタル信号処理の中で，FFT信号処理は多くの応用で使われます．FFTはいろいろな本で説明されているので，すでに皆さんはある程度(いやかなりかもしれない)の知識があると思います．FFTは高速に離散フーリエ変換を行うアルゴリズムのことです．

実際の時間軸の処理から，周波数領域での処理を可能とするための橋渡しをします．すでに第5章にも説明しましたが，信号を正弦波の集合として見ることによって，さまざまな処理を行うものです．言い換えればスペクトル解析です．

もともとの離散フーリエ変換(DFT)は，かなりの数の掛け算を含み，かなり重い処理で，実時間処理として使うには実装の可能性は見えませんでした．これを，三角関数の周期性を利用して，掛け算を減らし，高速に計算できるように考えられたのが高速フーリエ変換＝FFTです．当然，周波数領域の信号を実時間信号に戻すための逆高速フーリエ変換＝IFFTもあります．DFTを式で書くと次のように表せます．

$$C_K = \frac{1}{N}\sum_{n=0}^{N-1} X_n \cdot e^{-i\frac{2\pi nK}{N}} \qquad (K = 0, 1, \cdots, N-1)$$

正弦波なので，当然位相があり，結果は複素数となります．逆のIDFTは次のように表せます．

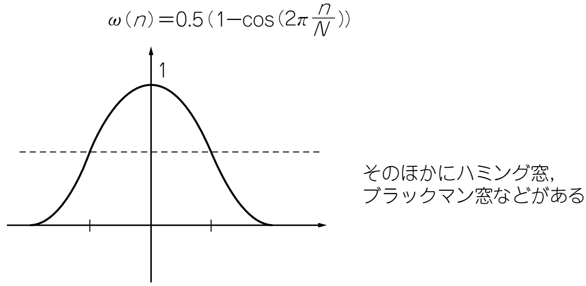

図6-1　窓関数(ここではハニング窓)

$$X_K = \sum_{n=0}^{N-1} C_n \cdot e^{i\frac{2\pi nK}{N}} \qquad (K = 0, 1, \cdots, N-1)$$

$1/N$は正規化のためですが，IDFTのところで行っても同じ結果です．あるいは両方に均等に分配し，それぞれ$1/\sqrt{N}$で割っても同じ結果になります．信号処理的にはIFFTのところで行ったほうが，受信側でノイズ成分を多く含むので，割り算を行うことによってS/N的には有利です．しかし，それだけ固定小数点のダイナミック・レンジを必要とします．実装の問題では，この固定小数点のダイナミック・レンジをどうするかを含めて決める必要があります．

多くのアプリケーションでは，FFTを使ってスペクトル解析をする場合が多いと思います．その場合は，信号のFFT処理された結果だけを使います．しかし，FFTを使い周波数面での信号処理をするための道具と考える場合は，処理された後でさらにフーリエ逆変換で時間軸の信号に戻す必要があります．2回のFFT処理が必要になり，処理時間は倍かかってしまいます．

6-1-2　境界の問題

フーリエ変換は無限時間の積分を行うものですが，それを一定時間に区切ったために，問題が発生します．

そこで，境界の不連続の影響を軽減するために，FFT処理をかける前に，**図6-1**に示すような窓関数による重み付けを行います(窓関数はほかにも何種類かある)．その辺の話は，第5章で説明したので参考にしてください．筆者はFFT処理を使ったフィルタ処理を中心に応用してきたので，次に述べる信号のオーバラップの問題から，主に上下対称なハニング窓を使ってきました．

6-1-3　信号のつながり

先ほど，窓関数をかけてからFFTを行うと書きました．これでは確かにスペクトル解析だけにこの結果を使う分には何も問題はありません．しかし，FFTを使ったフィルタリングなどの場合は，周波数軸で信号の操作を行った後にIFFTを行う必要があります．また，逆FFTされた時間軸の波形には，窓関数の影響が強く出ると思われます．FFTの周期に比例したひずみが発生し都合の悪いことになってしまいます．

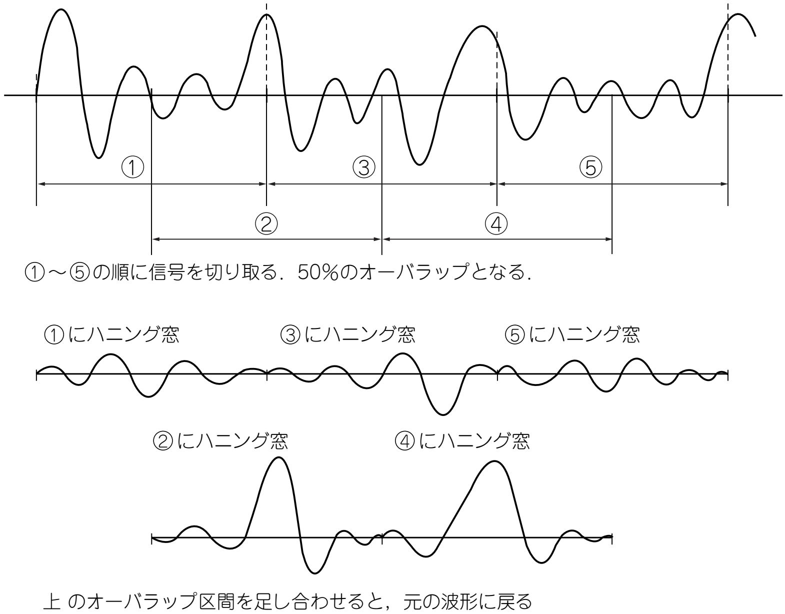

図6-2　FFT信号のオーバラップ

これを避けるために，**図6-2**のように前後のFFTと，信号のオーバラップを行います．一般的には窓関数の形に合わせて50％のオーバラップをすることが多いと思います．窓関数を，FFTブロックの両端のわずかな部分しかかけない場合もあり，そのときはオーバラップもそれに合わせた長さになります．もし，窓関数がハニング窓のように波形の中央で真中で上下対称波形の場合で50％のオーバラップをした場合は，窓関数自体がきれいなフェードIN，フェードOUTの特性をしており，境界できれいにつながります．

筆者は，ハニング窓による半分のオーバラップをすることが多いのですが，オーバラップの長さにはもちろん規定はありません．文献によってはもっと短い区間を行っている例も示されています．

このFFTとIFFTを組み合わせて使う信号処理は，とても強力です．直接周波数軸上で信号処理を行えますから，どんな周波数処理や補正もできてしまいます．

6-1-4　波形の広がり

周波数軸でのフィルタリングのように，FFTした結果をいろいろ加工しそれをIFFTして元の時間領域に戻す場合，波形が元あった時間領域をはみ出る場合があります．たとえば，信号のある周波数特性の遅延特性を補正しようとした場合，信号の遅延時間を変えるわけですから，時間軸に戻したとき元のブロックを飛び出してしまうことは充分考えられることです．

その場合，**図6-3**に示すように，FFTを行う信号の後半にゼロの区間を挿入し，そこを含めてFFT，IFFTの計算を行うようにします．IFFTした波形の結果が前後に広がっても，そのゼロを挿