5-17 FFT：FFTの基礎

フランスの貴族であり技術者であったフーリエが発表した理論は、今日、高速フーリエ変換（FFT）と呼ばれるアルゴリズムに形を変え、デジタル信号処理の基礎的な方法の1つになっている。FFTによって開かれるオーディオ処理の可能性は非常に広く大きい。音響合成から始まって、フィルター処理、エフェクト処理、空間処理へと展開してきたChapter 5の内容は、FFTの観点から見直せば、また新しい方向で展開していくことができるだろう。だがFFTを使いこなすためには単にオブジェクトの使い方を知っているだけでは十分ではなく、その背景にある基本的な理論を理解してはじめて自分の目的に応じた使い方ができるようになる。そのためには、直接に音とは関係ないように見える数学上の話題にもある程度触れざるを得ない。ここではまずFFTの基礎について、その基本的な考え方をできるだけ平易な言葉で解説していくことにする。

フーリエ変換で何ができるのか？

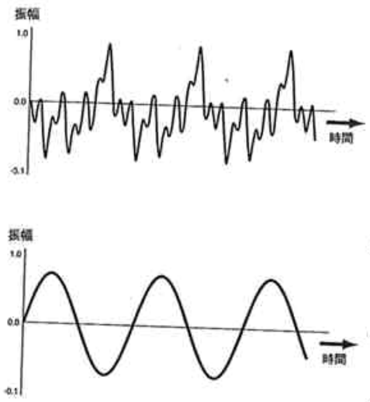
フーリエの理論について説明する前に、音楽あるいはサウンド・メイキングの目的にとってフーリエ変換でいったい何ができるのかという点を押さえておきたい。

音とは音源となる物体が振動し、それが空気中を波として伝播して、耳の鼓膜を振動させる物理現象である。振動や波は時間とともに変化する。したがって音を考えるとき、それを振動とその時間的な変化という観点から見るのが最も自然だろう。横軸に時間、縦軸に振幅をとってこれを表すと、音の時間領域表現が得られる。

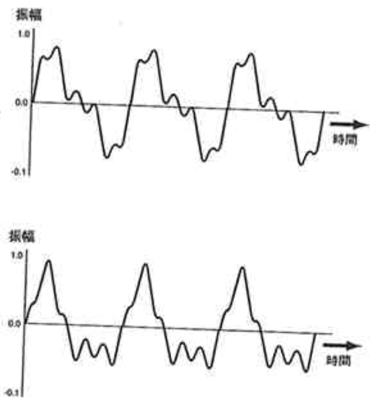
■5-17-1　音の時間領域表現

音の時間領域表現において描かれるグラフは音の波形に他ならない。異なった音色の2つの音があるとすると、5-17-2のようにその波形は必ず異なっているはずだ。しかし5-17-3の2つの音は波形という点ではまったく異なっているにもかかわらず、そのサウンドを聞くと同じ音色に聞こえる。つまり、音色が異なれば必ず波形が異なるとは言えても、その逆の波形が異なれば必ず音色も異なるとは言えないのである。（これは主として位相の問題になる）。

■5-17-2　異なった音色では波形が異なる例



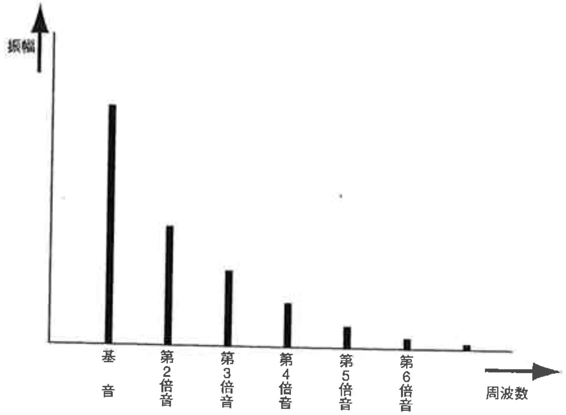
■5-17-3　波形が異なっているのに音色が同じ例



では、5-17-3の2つの音では何が同じなのだろうか？答えは部分音の構成バランスが同じだということになるだろう。

音は実験的に作り出された純粋なサイン波を除いて、すべて部分音を含んだ複合音である。1つの音としてわれわれが聞いているものには複数の部分音が混じっており、それぞれの部分音は単純なサイン波で表すことができる。ここに音を表現するもう1つの方法がある。横軸に部分音の周波数、縦軸に部分音の振幅をとってこれを表すと、音の周波数領域表現が得られる。

■5-17-4　音の周波数領域表現

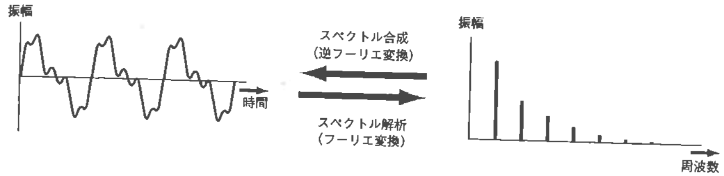


音の周波数領域表現において描かれるグラフは、スペクトル（あるいはスペクトラム）と呼ばれる。ちょうどプリズムによって光に含まれるさまざまな波長を分解するように、音のスペクトルも音の内部構造を分解して示してくれる。部分音は倍音とも呼ばれ、最も周波数の小さい倍音を特に基音と言う。基音とその他の倍音の周波数の関係が整数で表せる場合、それらを整数倍音と呼び、整数で表せない場合非整数倍音と呼ぶ。また倍音は周波数の低い方から順に第1、第2、第3……第n倍音と数えることができる。音の音色はこうした部分音の周波数関係、それぞれの振幅の大きさによって大きく決定されている。

さて、音の時間領域表現を周波数領域表現に変換すること、つまり波形をスペクトルに変換することをスペクトル解析と呼ぶ。スペクトル解析は、時間とともに刻々と変化する音のスナップショットを撮り、その内部構造を部分音に分解して示すことだ。これから紹介するフーリエ変換は、まさにこのスペクトル解析を行う技術なのである。

波形をスペクトルに変換する方法が分かれば、逆にスペクトルを波形に変換する方法もつかんだことになる。この逆方向の変換をスペクトル合成と呼び、それを行うための方法が逆フーリエ変換である。

■5-17-5　スペクトル解析とスペクトル合成



フーリエ変換および逆フーリエ変換が、音楽あるいはサウンド・メイキングの目的にとってどのような意義を持っているかは、もう理解してもらえるだろう。フーリエ変換は音楽の素材になるサウンドを分析し、それがどういった内部構造になっているのかを示してくれる。われわれは、この分析結果を音楽制作に存分に利用することができる。さらに、分析結果を元にそれを再現したり、自由に変更するといった極めて魅力的な領域が開ける。例えばローパス・フィルターは音の高域成分をカットするものだった。これを実現するためにディレイの原理を応用してデジタル・フィルターを作り、デジタル信号処理を行ってきた。しかし、この方法はある意味で間接的な方法である。目的は高域成分をカットすることであって、ローパス・フィルターはそのための間接的な手段になる。だが、スペクトルをもし自由にコントロールすることができれば、われわれは音にダイレクトに関わり、直接、高域成分をカットすることができるわけだ。しかも現実のローパス・フィルターでは構造上ありえないフィルター特性であっても、スペクトルを直接コントロールすることで実現できるはずだ。

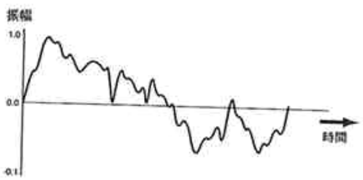
つまりスペクトル解析とスペクトル合成は、音そのものの内部構造にメスを入れ、その構造自体をコントロールする画期的な方法論になる。これを実現してくれるのがフーリエ変換および逆フーリエ変換だ。フーリエの理論は音楽を志す者にとっては決して分かりやすいものではない。指数関数、三角関数、複素数、積分といったおよそ音楽とは無関係に見える数学上の議論に触れざるを得ない。また音についての物理学的知識も必要になる。しかし、この珠玉の宝庫を前にして門外漢だと決め込んであきらめるのはあまりにもったいないだろう。本書ではできるだけ数学上の難解な議論は避けて、分かりやすい図式的説明を試みたい。フーリエの基本的な考え方が理解できれば、FFT関連オブジェクトも自分の創意工夫で使用していくことができるだろう。

フーリエ級数からFFTへ

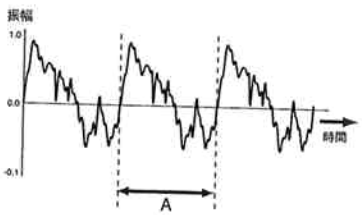
19世紀の初めにフーリエが提出した理論は、"どんなに複雑な波形も、周期的な波形であれば、周波数、振幅、位相の異なったサイン波の組み合わせに分解できる"と主張するものだった。数式を使わずにこの考え方をもう少し詳しく見てみよう。

まず、フーリエは、どんなに複雑な波形も、長い時間のスパンで見れば"周期的な波形"つまり同じ形が繰り返される波形であると"仮定"する。例えば、現実に5-17-6のような波形が与えられたとしよう。この波形をどう眺めてみても、ここから周期的な波形の繰り返しを見つけることはできない。通常これは非周期的波形と呼ばれる。

■5-17-6　繰り返しが見つけられない波形の例



■5-17-7　周期的だと推測される波形



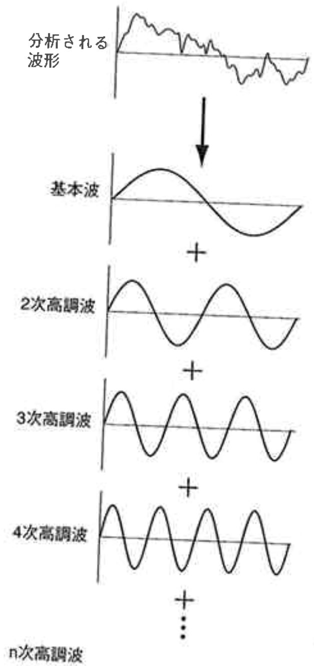
しかし、もう少し長い時間この波形の動きを観察してみたところ、5-17-7のような波形が得られたとしよう。さきほど非周期的だと思っていた波形が繰り返され、実は周期的な波形の一部分になっているように見える。つまり、この直後にもAの波形が繰り返され、それが永遠に連続していくと推測される。もちろん、この推測は裏切られるかもしれない。しかし、仮に裏切られたとしてもさらに長い時間波形を観察すれば、より大きな繰り返しの単位を見つけることができる可能性がある。このようにフーリエはまず与えられた現実の波形が周期的な波形だと仮定することから始める。この推測が現実には間違っていても構わない。あくまでも数学上の仮定なのである。

次に、周期的だと仮定された波形が"周波数、振幅、位相の異なったサイン波の組み合わせに分解できる"ことについて説明しよう。

フーリエは、先に例に上げた周期的波形の1周期（5-17-7のAの部分）を同じ周期のサイン波に置き換え、これを基本波と考える。そして、元の波形は、その基本波の周波数（基本周波数と呼ばれる）の整数倍の周波数を持つサイン波（高調波と呼ばれる）を5-17-8のように足し合わせたものだと考えたのである。なぜ整数倍なのかと不思議に感じるかもしれないが、与えられた任意の波形を周期的波形と仮定する以上、これは当然なのだ。なぜならもし非整数倍の周波数を持つサイン波が含まれていれば、それは繰り返しの度に基本波の周期とずれていき、結果として周期的波形という仮定に矛盾するからである。

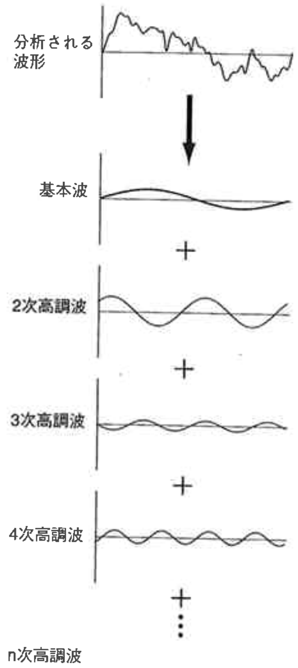
ちなみに基本周波数の2倍の周波数を持つ高調波を2次高調波、3倍であれば3次高調波……n倍であればn次高調波と呼ぶ。基本波と高調波の関係は、ちょうど基音と整数倍音の関係と同じだ。

■5-17-8　周期的だと仮定された波形に含まれるサイン波



ただし、5-17-8はフーリエの考え方を正確には反映していない。なぜなら基本波、高調波それぞれの振幅と位相を無視して描いているからだ。フーリエは、これら振幅と位相が5-17-9のようにそれぞれ異なっていると考えた。

■5-17-9　振幅と位相の異なったサイン波に分解



こうして、どんなに複雑な波形も、いったん周期的な波形だと仮定すれば、それは基本波と高調波の組み合わせ、つまり周波数、振幅、位相の異なったサイン波を足し合わせたものだと考えることができる。

これを数学的に表現したものをフーリエ級数と呼ぶ。つまりあらゆる波形（もちろん音波を含む）はフーリエ級数として数式で表現できることになる。

ところで、各サイン波の周波数は基本波の周波数が決まってしまえばあとはそれを整数倍していけばよいのだが、振幅と位相については分からない。そこで与えられた波形に含まれるすべてのサイン波について振幅と位相を分析する方法が必要になる。これこそがフーリエ変換（Fourier Transform＝FT）と呼ばれるものなのだ。フーリエの時代にはフーリエ変換は気の遠くなるほど膨大な手計算によって行われた。19世紀の終わりから20世紀初頭にかけて機械式のフーリエ解析機が発明されたが一般的な実用性とはほど遠いものだった。しかし、20世紀の中頃になってコンピューターが登場すると状況は一変する。

サンプリングされた有限個のデジタル信号に対するフーリエ変換は、離散フーリエ変換（Discrete Fourier Transform＝DFT）と呼ばれる。フーリエが想定した周期的な波形はつなぎ目のない連続的なアナログ量であり、しかも永遠に繰り返されるものだったが、コンピューターが扱うのは離散的なデジタル信号になる。離散フーリエ変換はコンピューター上に実装されたデジタル式のフーリエ変換だと言える。

しかし、手計算からコンピューターによる計算に代わったとは言え、離散フーリエ変換は膨大な計算量を必要とする。そこでいかに計算回数を少なくして、計算スピードを速くするかというプログラム上の工夫が求められるようになる。1965年に発表された高速フーリエ変換（Fast Fourier Transform＝FFT）は、魔法のようなプログラミング上のトリックを用いることで飛躍的に計算回数を減らすことに成功した。

したがってFFTはフーリエ変換に取って代わる新しい理論などではない。それはフーリエ変換をデジタル信号に当てはめた離散フーリエ変換を、さらに高速化する1つのアルゴリズムなのである。しかし、FFTのおかげでフーリエ変換はコンピューター上でリアルタイムに実行されるまでになり、多方面で実用化されている。そして、今われわれはFFTを音楽に応用しようとしているわけだ。

フーリエ変換と複素数

フーリエ変換やFFTを理論的に詳しく解説することは本書の範囲を超えている。ここでの目的は、フーリエの考え方の骨子、MSPにおけるFFT関連オブジェクトの背後にある考え方のエッセンスを理解することであり、最終的にはそれらによって開けるオーディオ処理プログラミングの可能性を探ることである。したがって数学上の用語や数

式はできるだけ使わず、図式的に説明することを心がけている。フーリエ変換やFFTについて詳しく知りたい読者は、デジタル信号処理関係の豊富な文献をあたってほしい。ただし、数学上の用語や数式を避けて説明しようとしても"複素数"という数学概念だけは触れざるを得ない。なぜならフーリエ変換やFFTはその計算過程で複素数を用いており、後述するfft~やifft~オブジェクトも複素数としてのシグナルを扱うからである。

通常、われわれが"数"と言われて思い浮かぶのは、整数、小数、分数などを含んだ実数(real number)であろう。これに対し虚数(imaginary number)と呼ばれる数学上の概念がある。それは2乗すると－1になる数として定義され、一般に"i"という記号で表される。

$$x^2 = -1$$
$$x = \sqrt{-1} = i$$

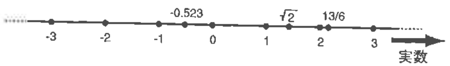
2乗して－1になる数など現実世界にはあてはめることができない。したがって想像上の数という意味で、これは虚数と呼ばれる。しかし、虚数の導入によって数の概念は拡張され、実数と虚数の両方を含んだ複素数(complex number)として次のように表される。

$$a + bi$$

ここでaとbは実数であり、もしb=0ならば複素数は単に実数を表現する。逆にa=0ならば純粋に虚数のみを表す。したがって複素数は実数と虚数の両方を含んだ最も広い意味の数だと言える。aの部分を複素数の実部、biの部分を虚部とも言う。

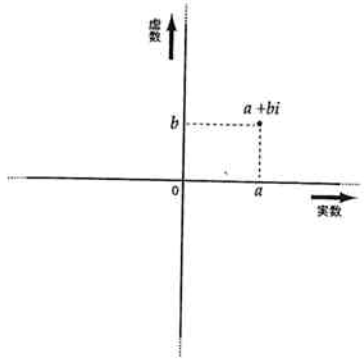
ところで実数は、それが整数であれ小数であれ分数であれ、また有理数あるいは無理数であれ、一直線に並ぶ数直線上の点として表すことができ、大小を比べることもできる。

■5-17-10　数直線上の実数

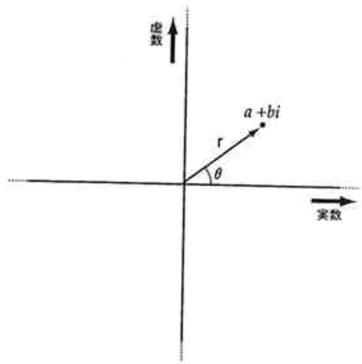


しかし虚数はこの数直線上に表すことのできない数である。このため複素数を表すためには、もう1つの次元を用意し、次のような2次元の平面上の点として表すことになる。この平面上で、任意の複素数は、aとbという直行座標あるいはデカルト座標(Cartesian coordinates)の値として特定することができる。

■5-17-11　直行座標で表した複素数



■5-17-12　極座標で表した複素数



さらに、平面上でこの同じ点を表す方法にはもう1つある。つまり、5-17-12のように複素数をベクトルの大きさrとその角度θという極座標(Polar coodinates)の値として特定する方法だ。

この極座標は、まさに$\sin\theta$や$\cos\theta$といった三角関数を表現する方法であり、rは振幅、θは位相の値として見ることができる。ここに来て複素数がフーリエ変換で用いられる三角関数と結びつくことが分かる。

これ以上の詳細な説明は控えるが、フーリエ変換はその計算過程で複素数を用いる。これは求めるべきサイン関数の振幅と位相を複素数の直交座標として計算していることになる。FFTも同様に複素数を用いる。コンピューター・プログラムの場合、複素数は実部と虚部に分けて計算され、計算過程で虚部が2乗されたときには実部に移す処理が行われる。したがってFFTの計算結果も、複素数の実部と虚部で出力されることになる。このポイントだけは押さえておいてほしい。

tableオブジェクト 画像解析 アナログ

FFTの基礎概念

高速フーリエ変換（FFT）について、まず次の点を確認しておこう。

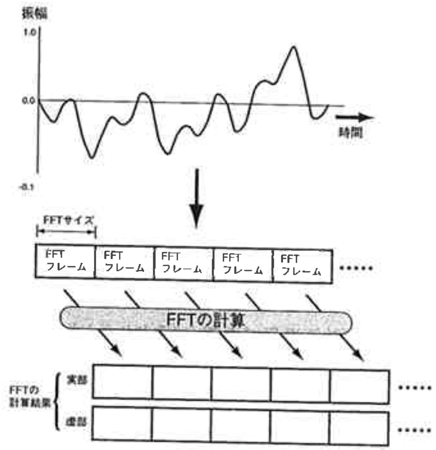
FFTはデジタル信号をスペクトル解析する
FFTが扱うデジタル信号は有限個のサンプルからなる
FFTは複素数を用いて計算を行う。このため実部と虚部は別々に計算される

■FFTフレームとFFTサイズ

FFTは時間とともに変化するデジタル信号を対象にし、それを短い時間のスライスとして切り取って分析を行う。この時間のスライスを映画のフレームになぞらえてFFTフレーム、その長さをFFTサイズと呼ぶ。FFTサイズはサンプル単位で指定され、必ず256、512、1024といった2のべき乗の数でなければならない。

FFTでは、FFTフレームをひとまとめにしてスペクトル解析を行い、それが済めば次のフレームを解析することを繰り返す。例えばFFTフレーム・サイズが512の場合、最初の512個分の入力サンプルを受け取るとFFTはそれを元に計算を行い、実部および虚部それぞれ512個のサンプルとして計算結果を出力する。

■5-17-13　FFTの実行



TRANS Max EXPRESS

■FFT周波数とFFTの計算結果

FFTサイズを512としたとき、このFFTの計算結果としての512個のサンプルはいったい何を示しているのだろう。まさしくこれがスペクトル解析結果の実部および虚部のリストなのである。

フーリエの考え方に従って、FFTはFFTフレームとして切り取った信号を周期的な波形であると仮定し、基本波、高調波に分解する。基本波の周波数はサンプリング・レート（サンプリング周波数）をFFTサイズで割ったものにほかならない。つまりサンプリング・レートが44.1kHzでFFTサイズが512だとすると、44100/512＝86.132……となり、およそ86.1Hzが基本波の周波数となる。これを特にFFT周波数と呼ぶ。高調波の周波数はFFT周波数の整数倍なので、2次高調波の周波数＝86.132……×2＝172.265……、3次高調波の周波数＝86.132……×3＝258.398……となりn次高調波の周波数は86.132……×nになる。

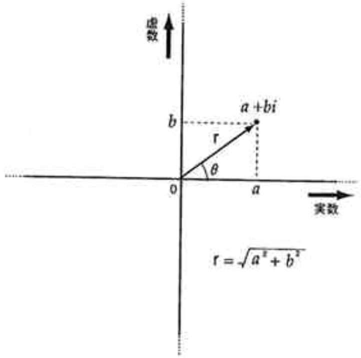
要するにこの場合、FFTは86.1Hzごとに周波数ポイントを合計512個定め、それぞれの周波数ポイントでのエネルギーの大きさを分析しているのだ。これらの周波数ポイントのことを周波数ビン（Frequency Bin）と呼ぶ。エネルギーとは各周波数ビンにおける入力信号の振幅と位相と時間要素を複素数で表現したものである。

■5-17-14　1つのFFTフレームのFFT計算結果

番号	周波数	実部	虚部
0	0.0	7.9041	0.0000
1	86.1	7.9815	-0.0166
2	172.2	8.2229	-0.0342
3	258.3	8.6594	-0.0540
4	344.5	9.3543	-0.0777
5	430.6	10.4300	-0.1083
6	516.7	12.1347	-0.1511
7	602.9	15.0376	-0.2183
8	689.0	20.7653	-0.3443
9	775.1	36.5152	-0.6807
10	861.3	238.5583	-4.9378
11	947.4	-46.6902	1.0622
·	·	·	·
·	·	·	·
·	·	·	·
500	43066.3	-46.6902	-1.0622
501	43152.5	238.5583	4.9378
502	43238.6	36.5152	0.6807
503	43324.7	20.7653	0.3443
504	43410.9	15.0376	0.2183
505	43497.0	12.1347	0.1511
506	43583.1	10.4300	0.1083
507	43669.3	9.3543	0.0777
508	43755.4	8.6594	0.0640
509	43841.5	8.2229	0.0342
510	43927.7	7.9815	0.0166
511	44013.8	6.5573	0.0000

この計算結果の実部と虚部を2乗して足し合わせ、その平方根をとると、振幅が計算できる。これは、先ほど説明した直交座標から極座標への変換であり、5-17-15のように考えることができる。そして、横軸に周波数、縦軸に振幅の大きさをとった平面に、この振幅をプロットするとスペクトルが描けることになる。

■5-17-15 直交座標から極座標への変換



■周波数精度と時間精度

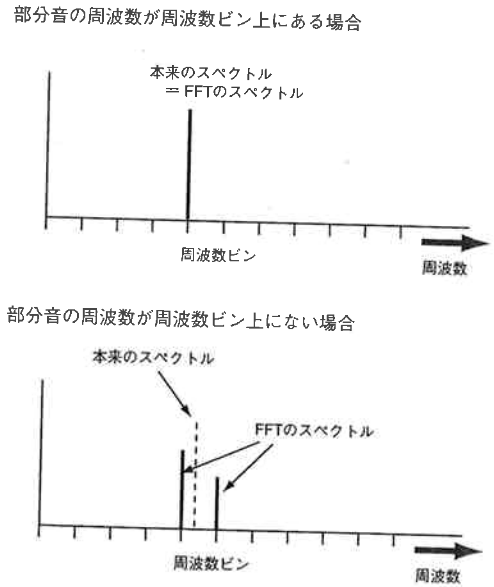
スペクトル解析において、周波数精度と時間精度は反比例する。FFTサイズをもっと大きくすると周波数ビンの数は多くなり、周波数の分割は細かくなる。ここではFFTサイズを512サンプルとして説明しているが、これを64倍して32,768サンプルとすると、周波数ビンの分割数も64倍になり、44,100/32,768＝1.3……Hzごとに解析を行うことができる。その結果スペクトル解析の周波数精度は向上する。

では、FFTサイズをどんどん大きくしていく方が好ましいのかと言えば、必ずしもそうではない。FFTサイズを大きくすることは、それだけ長い時間のスペクトルの変化を合わせることになり、スペクトルの細かな時間的変化をつかむことができなくなるからだ。つまり、FFTサイズを大きくしていけば、スペクトル解析の時間精度は下がっていく。

現実の音は時間とともに刻々と変化する。したがってスペクトルも刻々と変化する。512サンプルのFFTフレームはおよそ11.6msecの時間幅を持っており、スペクトルにはこの時間内に生じたすべての周波数が反映される。もしFFTサイズを32,768サンプルにすれば、1フレームの時間幅は0.7秒以上の大きなものになる。この0.7秒の間にはさまざまな音の変化が起こり得るだろう。例えば、歌手の歌声のスペクトル解析を行っている場合、この0.7秒間の中でメロディが変化したり、ビブラートなどでピッチが大きく変化しているかもしれない。得られるスペクトルはこれらの変化による周波数をすべて反映したものになってしまう。ある瞬間のスペクトルが欲しい場合にはFFTサイズはできるだけ小さい方が望ましい。結局、周波数精度を優先させるか、時間精度を優先させるかは、目的によって使い分ける必要があるだろう。

周波数精度については、もう1つの問題がある。音は複数の部分音で構成され、それぞれの部分音は単純なサイン波でできている以上、すべての部分音のスペクトルは1本の直線として表されるはずだ。もし部分音の周波数がちょうど周波数ビン上にある場合、FFTは最も満足のいく形で正確にスペクトル解析を行い、まさに部分音のスペクトルは1本の直線として表される。

■5-17-16 周波数精度とスペクトルの分散



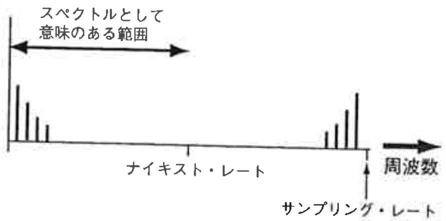
しかし、周波数ビンの間隔は、サンプリング・レートをFFTサイズで割ったものであり、分析する対象のデジタル信号とは無関係に決まってしまう。もし部分音の周波数が周波数ビン上にない場合、得られるスペクトルは1本の直線ではなく、複数の周波数ビンに分散した複数の直線になってしまう。これは避けられないことで、FFTを任意の周波数を含むデジタル信号に適用した場合、その結果は実際のスペクトルの近似となり、スペクトル解析は正確にはスペクトル推定と呼ぶべきものだと言える。もちろんFFTサイズを大きくしていけば周波数ビンの間隔は狭くなり、分析結果は実際のスペクトルに近づいて行くのだが、それでも近似であることには変わりはない。

■ナイキスト・レートによる折り返し

FFTの計算結果リストを見ていると面白いことに気づく。例えば実部について見ると、基本波86.1Hzのエネルギーと510次高調波43927.7Hzのエネルギーが同じ7.9815になっている。例えば10次高調波と501次高調波についても同様だ。つまりこのリストは、一番最初の数値と最後の数値を除けば、真ん中に鏡を置いたように対照になっているのである。虚部についても同様に真ん中に鏡を置いたように対照になっているが、さらに符号も逆になっている。

このようにFFTの結果リストは中央の周波数ビン(ここでは257番目)を境にして折り返される。これは、デジタル・オーディオが、サンプリング・レートの1/2(ナイキスト・レート)の周波数までしか表すことができないというサンプリング定理に対応している。つまりサンプリング・レートが44.1kHzの場合、原理的に、デジタル・オーディオの周波数は、22.05kHzまでしか扱えない。FFTの結果で43927.7Hzのエネルギーがいくらであろうと、これは意味を持たない。結局、スペクトルとして意味があるのは前半部分だけということになる。

■5-17-17 ナイキスト・レートとスペクトルの折り返し

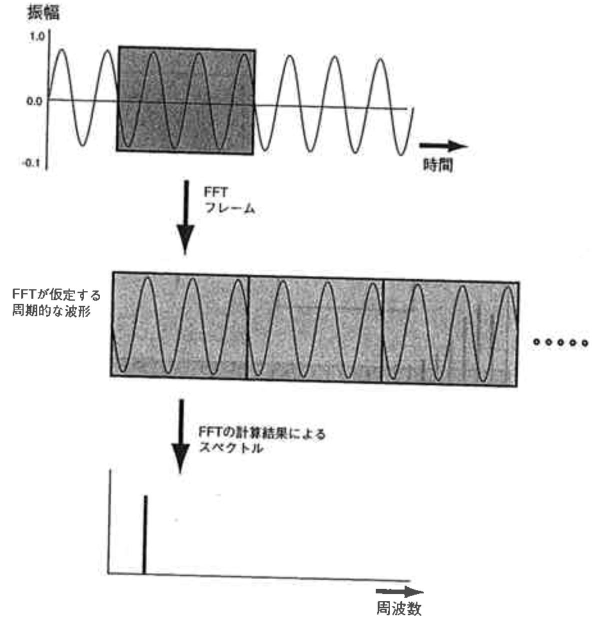


■窓関数の意義

FFTではFFTフレームとして切り出されたデジタル・オーディオ信号を周期的な波形だと仮定することはこれまでも説明した。しかし、ここに1つの問題が発生する。

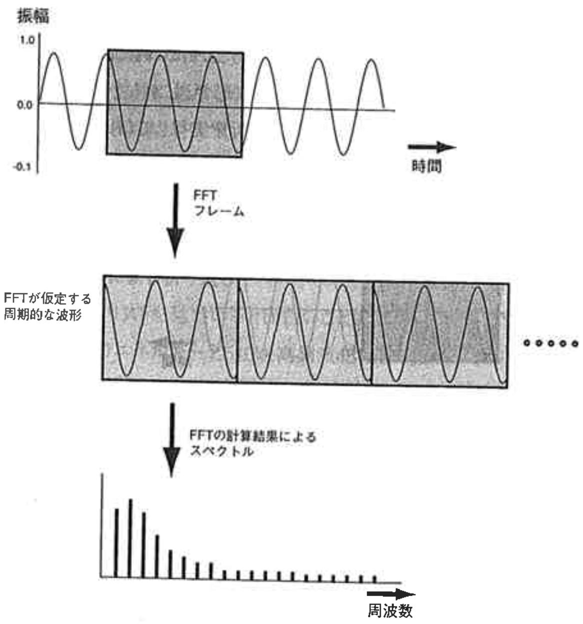
P652の「周波数精度と時間精度」の項で説明したように、例えば5-17-18のように分析対象となるデジタル信号が、そもそも周期的な波形で、しかもその周波数がFFT周波数の整数倍ならば、FFTは見事にスペクトル解析を実行してくれる。

■5-17-18 スペクトル解析がうまく成功する例



しかし、5-17-19のように分析されるデジタル・オーディオの周波数がFFT周波数の整数倍ではない場合、あるいはそもそも周期的な波形ではない場合、それでもFFTは切り出されたフレームを周期的な波形だと見なす。その結果、FFTが仮定する周期的な波形はおよそ元の波形とは異なったものとなり、分析結果はまったく誤ったものになってしまうだろう。

■5-17-19　スペクトル解析が失敗する例



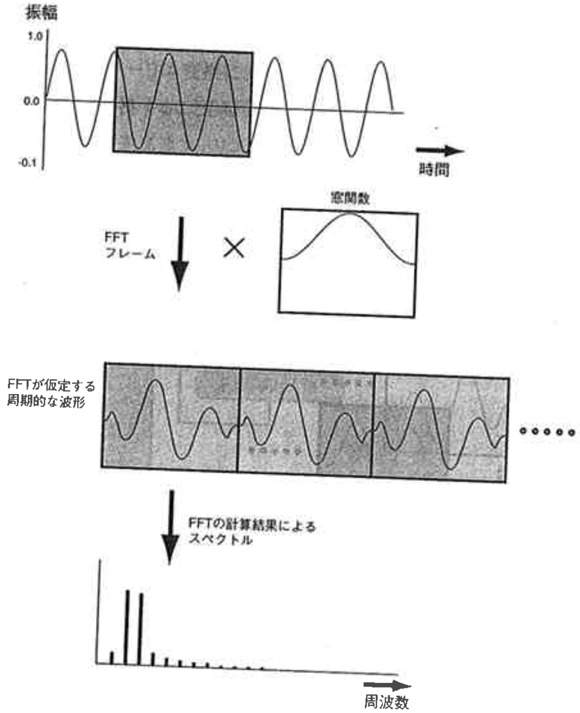
上の図を眺めていると、問題はFFTフレームの両端の部分がうまくつながらない点だということが分かる。ここで波形の不連続が生じるのだが、FFTは分析すべき本来の波形がそうした不連続を最初から持っているように見なしてしまう。不連続な波形はノコギリ波のように高次の周波数成分をたくさん含む。その結果、スペクトルには本来の波形には含まれていない周波数成分が多数現れてしまうことになる。

そこで使われるテクニックが窓関数と呼ばれるものだ。これは、ベル型などのエンベロープをFFTフレームに付けて、フレームの両端で振幅が必ず0になるようにするテクニックである。あるいは、FFTフレームごとにフェード・イン、フェード・アウトすると考えてもよい。エンベロープを付けるというのは、FFTフレームと窓関数の個々のサンプル同士を掛け合わせることである。窓関数という言葉は、釣り鐘型や三角型の窓を通し

てFFTフレームを見るというイメージで付けられている。

もちろん、5-17-20のように窓関数を使ってもFFTの分析結果がまったく正確になったとは言えない。また窓をかけること自体によって解析結果に歪みという副作用ももたらされる。しかし、それでも使わないときに比べて明らかに結果は改善される。

■5-17-20　窓関数を掛ける

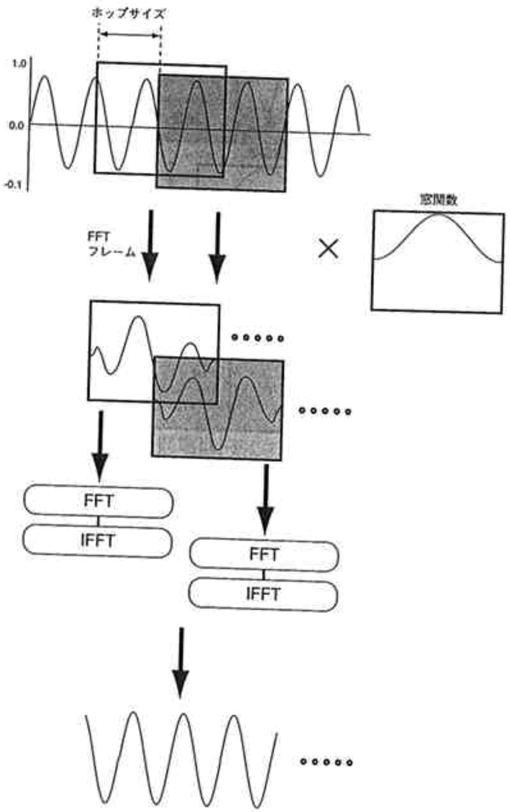


代表的な窓関数には、その窓の形により、"矩形(Square)" "ハミング(Hamming)" "ハニング(Hanning)" "ブラックマン(Blackman)"などがある。

■FFTのオーバーラッピング

窓関数を掛け合わすことで任意の波形に対してFFTは有効になる。スペクトル解析という点だけならこれでもよいが、次に説明する逆フーリエ変換でスペクトル合成を行うことを考えるとまだ問題が残る。逆フーリエ変換によって得られる波形は、元々フレームごとに窓関数を掛け合わせている結果、断続的なものになるだろう。これを避けるためにもう1つ別にFFTを行い、それを時間的にずらせてオーバーラップさせるテクニックが使われる。要するに"間を埋める"処理だと考えればよい。

■5-17-21　FFTのオーバーラッピングの考え方



オーバーラップさせるFFTの数は通常2ないし4が多い。これ以上にすると、窓関数による解析歪みの影響が大きくなる。オーバーラップの時間的なずれの大きさをホップサイズと言う。通常、ホップサイズは、FFTサイズをオーバーラップの数で割ったもの(2つのFFTによるオーバーラップの場合、FFTサイズの1/2)になる。これにより断続性は軽減され合成後の波形は自然なものになる。

■IFFT

フーリエ変換は信号の時間領域表現を周波数領域表現に変換するものであり、言い換えれば波形をスペクトルに変換することである。これが実現すると、数学的な手続きを逆にすれば、フーリエ変換によって得られるスペクトルを元に周波数領域表現を時間領域表現に変換することも可能になる。これを逆フーリエ変換と言い、FFTと逆の変換を行うことを逆高速フーリエ変換(Inverse Fast Fourier Transform＝IFFT)と呼ぶ。

■5-17-22　IFFTの実行