

スペクトラム解析

川村 新

3-1：高速フーリエ変換FFT/逆FFT

収録フォルダ：15_1_FFT_Through

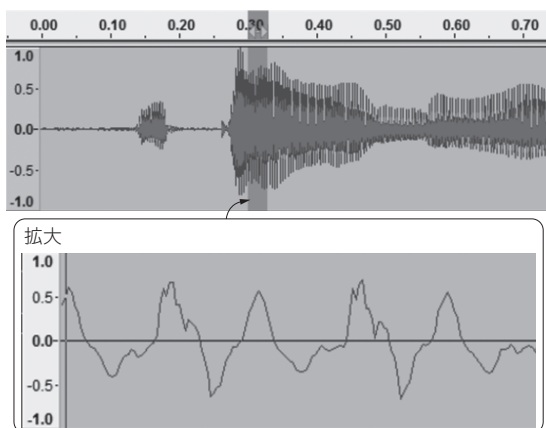


図1 音声の波形は短時間なら規則正しい繰り返しパターンになる

入力wavファイルに対して高速フーリエ変換(FFT: Fast Fourier Transform)を適用します。さらに逆FFT(IFFT: Inverse FFT)により、元の信号に復元します。

FFTによるプログラムができると、音声信号処理の幅が一気に広がります。

● 仕組み

▶ 周波数解析の基本は離散フーリエ変換

音声信号処理の主役級の周波数解析法は、離散フーリエ変換(DFT: Discrete Fourier Transform)です。 N 個の入力信号 $s(t)$ ($t = 0, 1, \dots, N-1$) に対するDFTは次のように定義されます。

$$X(k) = \sum_{t=0}^{N-1} s(t) \exp\left(-j \frac{2\pi k}{N} t\right) \dots\dots\dots (1)$$

ここで、 j は虚数単位です。また、 k は周波数番号を表します。 k 番目の周波数と実際の周波数 F [Hz] は、次のように関係しています。

$$\frac{k}{N} = \frac{F}{F_s} \dots\dots\dots (2)$$

ここで、 F_s はサンプリング周波数です。 N が大きいほど、周波数分解能は高くなりますが、その分、DFTに使用する $s(t)$ のサンプル数は多くなります。

分析結果の $X(k)$ から元の信号 $s(t)$ を得るには、次の逆DFT(IDFT: Inverse DFT)を利用します。

$$s(t) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left(j \frac{2\pi k}{N} t\right) \dots\dots\dots (3)$$

▶ 離散フーリエ変換を高速に行う

FFTは離散フーリエ変換の高速算法です。よって、FFTの結果とDFTの結果は同じです。また、IFFTもIDFTの高速算法であり、結果は等しくなります。

FFTアルゴリズムの詳細は他の書籍に譲ります。音声に対するFFT分析は、フレームと呼ばれる短時間の区間ごとに順次実行します。これは、長時間では激しく変化する音声も、短時間(30ms程度)ならば特性がほとんど変動しないためです(図1)。

フレーム内の $s(t)$ の始点と終点の値は通常異なりますが、このとき分析誤差が増大することが知られています。分析誤差を少なくするために、窓関数と呼ばれる関数を掛けて、フレーム内の $s(t)$ の両端を滑らかにゼロにします。

また、FFT分析は、フレームの長さを半分ずつシフトして実施します。IFFTで出力を合成する場合、フレームが半分重なりますが、この部分は信号を加算することで出力を作成します。

● プログラム

FFTのプログラムをリスト1に示します。

今回のFFTは、 $s(t)$ の512サンプルを1フレームとして実行します。サンプリング周波数が16kHzなので、フレームの長さは32msに相当します。

FFTの結果は複素数になるので、実部をxr、虚部をxiの配列で表現しています。

● 実行結果

実行結果を図2に示します。入力波形と出力波形の