

フーリエ変換

素人がフーリエ変換についてあーだこーだ語るの荷が重いので、具体的な計算方法や処理の仕方については、解説サイトが結構な数ありますので、ググってそちらを参照してください。ここでは概略と要点の解説にとどめておきます。

フーリエ変換を行うと、以下のイメージのように、時間軸での物理量データを周波数軸の強弱データに変換することができます。どの周波数帯の成分が強く現れているのか、一目瞭然になります。

離散フーリエ変換 (DFT) と高速フーリエ変換 (FFT)

本来のフーリエ変換はアナログ波の単一パルスに対して行う処理です。しかし、コンピュータ上で扱えるデータは標本化されたサンプリングデータです。これは連続していない飛び飛びの (= 離散している) データです。コンピュータ上で処理する場合は、離散フーリエ変換 (DFT: Discrete Fourier Transform) を用います。

DFT は計算量が多いため非常に時間がかかります。そこで実際には高速フーリエ変換 (FFT: Fast Fourier Transform) を使用することになります。FFT と DFT は基本的に同じ処理なので、計算結果は (誤差を除けば) 同じになります。唯一の制限が計算できる波形の長さが 2 の n 乗に限定されてしまいますが、DFT に比べて非常に高速に演算が行えるようになります。

窓関数

フーリエ変換は暗黙のうちに「処理する波形を一周期とした、無限に続く波形」と見なして計算を行っています。この時、始点と終点の値があまりにもかけ離れていたりするなど、音としての連続性に不備があると、本来は存在しない変化が発生してしまい、都合がよろしくありません。

この対策として、自然な形で始点と終点がゼロになるように補正をします。この補正用関数は窓関数 (Window Function) と呼ばれています。(・・・厳密に言うと嘘っパチな解説かもしれませんが、でも、この認識で問題ないと思います。) 目的によって様々な窓関数が使われているそうです。

いろいろテストしてみたのですが、窓関数あり・なしで得られた結果を実際に聞いてみても、筆者には違いがよくわかりませんでした。窓関数を使うと音が丸くなったような感じがする程度の違いは感じましたが・・・今回は一応クリティカルな場合を想定してハミング窓を採用しました。さらにちなみに窓関数なし (= なにもしない窓関数) を「矩形窓」と言うそうです。

それなりのキーワードは書きましたので、興味のある方はググってくださいませ ^.^;

この辺りは「そーゆーもんだ」とあまり深く考えずに居たほうが良いと思います。筆者自身も厳密に理解しているわけではありませんし、はっきりいってブラックボックスと割りきって使いました。実際の運用も、露本伊佐男氏の ExcelVBA 用高速 FFT ルーチンの解説 を理解するだけで必要十分だと思います。