

6 フーリエ変換

正弦波を重ね合わせるといろいろな音色の音を作れた。逆に、音をいくつかの正弦波に分解することもできる。それを可能にする技術としてフーリエ変換がある。本講義で扱うデジタル信号に対しては、離散フーリエ変換という技術がある。fft という関数で計算できる。さっそく fft を試してみる。

ソースコード 4: fft の使用例

```

1 >> fs=100;
2 >> t=0:1/fs:6;
3 >> y=sin(2*pi*15*t)+sin(2*pi*40*t);
4 >> f=fft(y,600);
5 >> f(10)
6
7 ans =
8
9     7.5656e-013 -3.0956e-013i    % fft の結果は複素数
10
11 >> s=abs(f);    % 複素数の「大きさ」を計算
12 >> plot(s)

```

信号 y は実数からなる。実数の信号をフーリエ変換すると複素数の結果が得られる。この例では、4行目で y の先頭から 600 点分をフーリエ変換している。(y 全体は、600 点ではないことに注意) フーリエ変換の結果の f の 10 番目の値を 5 行目で確認している。確かに複素数になっていることがわかる。フーリエ変換の結果をスペクトルという。スペクトルの絶対値をとったものを振幅スペクトルという。この例の場合、プロットすると、真ん中の 300 点のところを中心に線対称の関係で、2 本ずつの線が観測される。この線の位置が信号の成分を表わす。

練習 17 ソースコード 4 の例の f の要素のどの点とどの点と同じ値になっているかを調べよ。

ソースコード 4 の振幅スペクトルをプロットすると、全部で 4 本の線が観測されたはずである。どの位置に線があるかを調べるには、プロット図を拡大する方法や、線の周辺だけをプロットしてみる方法などがある。

しかし、図から観測するのではなく、プログラミングでも線のある場所を調べることができる。

```

1 >> a=[5 1 0 2 6 3]
2
3 a =
4
5     5     1     0     2     6     3
6
7 >> a>4
8
9 ans =
10
11     1     0     0     0     1     0
12
13 >> a(a>4)
14
15 ans =
16
17     5     6
18
19 >> find(a>4)
20
21 ans =
22
23     1     5
24
25 >>

```

この例では、7 行目でベクトルに対して比較演算子である $>$ を適用している。この場合は、 a のそれぞれの要素に対し、4 と比較して、4 より大きければ、1 をそうでなければ 0 を返している。(11 行目: 1 は論理値の真を表わし、0 は偽を表わす)

13行目は MATLAB 独特の書き方である。a() はベクトル a の要素を指定しているのであるが、ここでは論理配列を引数としている。論理配列を引数とした場合、真のインデックスの値だけを出力する。この場合、1番目と5番目だけが真なので、1番目の値 5 と 5番目の値 6 が表示される。

値を直接出力するのではなく、インデックスを調べる際には、find という関数を利用する (19行目)。find の使い方は help で各自調べること。

練習 18 ソースコード 4 の線があるインデックスを find 関数を使って調べるプログラムを書け。

振幅スペクトルの横軸は周波数に対応している。ソースコード 4 の例では、サンプリング周波数 (2) Hz に対し、600 点のフーリエ変換を行っている。600 点で (2) Hz を表しているの、1 点では、(2) Hz を 600 で割った周波数を表す。

練習 19 下線部 (2) を埋めよ。また、上記の説明を踏まえて、ソースコード 4 の結果の線の部分がどの周波数を表すかを考えてみよ。

練習 20 ソースコード 4 の s の s(1) から s(301) までをプロットせよ。ただし、x 軸の単位が Hz になるように工夫せよ。

7 窓関数

フーリエ変換は信号を正弦波に分解できる、と書いたが、実はその説明では不十分である。厳密には、周期性の信号でない正弦波には分解できない。離散フーリエ変換では、フーリエ変換する範囲を有限としているが、その範囲の外側では、範囲の中と同じ変化が繰り返されていると仮定している。つまり、ソースコード 4 の場合、y(1) から y(600) までの範囲での変化が、y(601) から先も続くとしている。実際、15Hz と 40Hz の正弦波は y(i) と y(i + 600) は同じ値となるので問題ない。

しかし、逆にいうと、次のような場合は問題が生じる。(y はソースコード 4 で作成したもの)

ソースコード 5: 周期と異なる点数での FFT

```
>> plot(abs(fft(y,599)))
>> plot(abs(fft(y,598)))
```

fft の長さが 600 点ではないので、最後のところで周期性が若干崩れる。そのため、y の成分に対応したインデックスだけでなく、その周辺の値も 0 より大きくなってしまっている。このような現象をリーク (leak, 漏れ) と呼ぶ。要するに、実際には含まれていない成分が含まれるという分析結果になってしまっている。

練習 21 ソースコード 4 の y に対し、600 以外でリークが生じない fft の長さを見つけてその条件で fft をかけて、その結果をプロットせよ。

信号に含まれている成分の周期があらかじめわかっているのであれば、fft の点数は、周期にあわせた点数にすればよい。しかし、fft は、信号にどのような成分が含まれているかわからないときに、それを知りたくて用いることが多い。したがって、ちょうどよい点数で fft をかけることは不可能に近い。

そこで、実際の信号を fft で分析するときには、窓関数を利用することが多い。窓関数を使うことで、fft をかける信号の両端の変位 (値) をなるべく 0 に近い値にし、擬似的に周期性を担保する。

窓関数の例としてハン窓 (ハニング窓) を見てみる。

```
>> w=hann(600);
>> plot(w)
```

このように中央部分は 1 で裾に向かってゆるやかに 0 になるように変化する。信号に窓関数をかけるといのは、窓関数で振幅変調をおこなうことである。

練習 22 ソースコード 4 の y の $y(1:599)$ に同じ長さのハン窓をかけて、その結果をプロットして観察せよ。(hann 関数の返却値は列ベクトルであることに留意せよ)

窓関数を用いることで、リークが減ることを確認する。

練習 23 ソースコード 5 を `fft` をかける前に窓関数かけることで、リークが減ることを確認せよ。窓関数による前処理の有無で、どのような変化が生じるかを観察し、考察せよ。

練習 24 周期性がありそうな音を録音し、その音を MATLAB に取り込め。取り込んだらプロットして波形を観察せよ。周期性がありそうなところを周期性が確認できるような範囲でプロットせよ (横軸の単位は時間になるように工夫せよ)。その範囲を 2 種類の条件で `fft` を用いて分析せよ。ただし、適切に窓関数による前処理をおこなうこと。この結果を用いて、対象とした音の成分について考察せよ。

8 音声のフレーム処理

フーリエ変換は `fft` の長さの間は、対象となる信号は周期的であると仮定している。したがって、対象となる信号の周期性や性質によって `fft` の長さを適切に選ばなければならない。例えば、120BPM (1 分間に 4 分音符が 120 回演奏される速さのこと) のメロディーの 4 分音符の音色を分析したい場合であれば、その音符の間ずっと全く同じ性質であったとしても $60s \div 120 = 0.5s$ なので、500ms 以内の長さで分析しなければならない。

また、`fft` が、分析対象となる区間に常にぴったり合うとは限らない。例えば、「おんせい」という単語の発声データでは、「お」「ん」「せ」「い」という音、それぞれの長さは違うし、どこからどこまでがどの音なのかも実ははっきりしない。このような対象に対しては、少しずつずらしながら分析することが一般的である。

長いデータを少しずつの区分に分割して処理するとき、この区分のことをフレーム (frame) とよぶことがある。MATLAB には、1 次元のデータをフレームに分割し、窓関数をかけて `fft` 処理することをまとめて処理する関数が用意されている。`spectrogram` である。

サンプリング周波数 8KHz で収録された音声データ (仮にここではファイル名を `xxx.wav` だとし、そのファイルは、MATLAB のカレントディレクトリのところにコピーしておいたとする) を `spectrogram` で処理する例である。

ソースコード 6: `spectrogram` を用いてフレームごとのスペクトルを求める例

```
>> [y, fs, nbits] = wavread('xxx.wav');
>> S=spectrogram(y,hann(256),128,256);
>> size(S)

ans =

    129    206

>>
```

最後の `size(S)` の答の 2 つ目 (上記では 206) は、ファイルによって変わる。

```
>> size([1 2; 3 4; 5 6])

ans =

     3     2

>>
```

`size` 関数はこのように、 3×2 の行列に対して、`[3,2]` を返却する。したがって、ソースコード 6 の `S` は 129 行 206 列の行列となっている。この 1 列目は `y` の最初の 256 点にハン窓をかけて `fft` した結果が含まれている。

ただし、`fft` の結果の後半は前半から計算できる (6 章では絶対値をとったものに関して前半と後半が線対称になると述べたが、複素数としては、対応する点が共役複素数となる) ので、前半の 129 点だけを含んでいる。したがって、列ベクトルの長さは 129 となっている。

ソースコード 6 の spectrogram の第 2 引数はフレームにける窓関数、第 3 引数はずらすときに重なるデータ点数、第 4 引数は fft の長さである。

spectrogram を用いて、フレーム処理して求めたスペクトルの様子を可視化することもできる。

```
>> spectrogram(y,hann(256),128,256,'yaxis')
```

表示されているように横軸が時間で縦軸が周波数である。赤い部分が強い周波数成分を表す。このような可視化の方法およびその図をスペクトログラムとよぶ。スペクトログラムでは、周波数成分の時間変化を見られる。

同じデータに対し、ずらし幅はそのまま fft の長さを増やしてみる。

```
>> figure(2)
>> spectrogram(y,hann(1024),128,256,'yaxis')
```

figure(2) は図を出力するウィンドウを指定している。何も指定しない場合は、figure(1) で指定するのと同じである。このようにすると同時に複数の図を表示できる。

練習 25 適当な音声 (単語か文を発声したもの)、もしくは、楽器音 (音楽でも構わないが、その場合は単独の楽器のみが使われているようなものがよい) のスペクトログラムを様々な fft の長さで表示し (ただし、ずらし幅は一定とする)、比較し、fft の長さで表示されるスペクトルの関係について考察せよ。(fft の長さは極端に長いものも試してみる方が考察しやすいだろう)

9 逆フーリエ変換

フーリエ変換は時間領域の信号を周波数領域に変換する計算である。その逆に、周波数領域の信号を時間領域に変換する計算が逆フーリエ変換である。MATLAB で fft の逆は ifft である。

スペクトログラムの結果得られたフレームごとのスペクトルは省略されている部分を追加 (復元) することによって ifft で時間波形を復元できる。

ソースコード 7: ifft を用いたスペクトログラムからのフレームの時間波形の復元

```
1 >> SL=S(:,10); % 10フレーム目を取り出す
2 >> SR=flipud(----(S(2:end-1,10))); % 複素共役を計算し、列ベクトルを上下反転させて
3 % スペクトログラムの右の部分を生成する
4 >> S10=[SL;SR]; % 列ベクトルを結合する
5 >> plot(ifft(S10)) % 逆フーリエ変換して表示する
```

練習 26 ソースコード 7 の下線部を埋めて完成させよ。下線部には共役複素数を計算する関数名が入る。

フレームをずらし幅を意識しながら足し合わせると (完全に同一にはならないが) 時間波形を復元できる。

ソースコード 8: フレーム表現の複素スペクトルからの時間波形の復元

```
1 >> fftlen=256;
2 >> noverlap=128;
3 >> slen=length(S);
4 >> ry=zeros(1,----*fftlen-(-----)*noverlap); % 復元される時間波形を格納するベクトルを予め作成
5 >> for i=1:slen, % インデ克斯をずらしながら復元した時間波形を足し合わせてゆく
6 SL=S(:,i);
7 SR=flipud(----(S(2:end-1,i)));
8 il=-----; % 開始インデクスを計算
9 ry(il:-----) = ry(il:-----)+ ifft([SL;SR]);
10 end
11 >> plot(ry)
```

練習 27 ソースコード 8 の下線部を埋めて完成させよ。

課題

以下の課題をパワーポイントファイルに作成し、授業支援システムのみ Navi の第 3 回目の「準備課題」のところでアップロードすること。パワーポイントファイルは、18 ポイント以上の大きさの文字を使って作成すること。締切は 10/3(日) 午後 8 時とします。自動で締め切ってしまいますので、十分に注意して下さい。音を生成・加工する課題については、音データもパワーポイントのスライドに貼り付けるようにすること。

1. 練習 17 を解答せよ。
2. 練習 18 を解答せよ。
3. 練習 19 を解答せよ。
4. 練習 21 を解答せよ。
5. 練習 24 を解答せよ。
6. この資料を読んで、わからなかった点があれば、わからなかった点が解消されるような回答が期待できるような具体的な質問を考えて書くこと。(これに関しては、スライドではなく、別のファイルに作成してアップロードせよ)