

19 任意の音の周波数変調

チャープ信号のように周波数を変化させることを掃引 (スィープ) ともいう。

任意の音 (録音した音) の周波数変調を行う方法を考えてみる。

式 (25) の考え方をを用いると、例えば、1 秒間で f_1 Hz の正弦波を f_2 Hz まで直線的に変化させる場合には、次の式になる。

$$\begin{aligned} \sin(2\pi f(t)) &= \sin\left\{2\pi\left(f_1 t + \frac{f_2 - f_1}{2} t^2\right)\right\} \\ &= \sin\left\{2\pi f_1\left(t + \frac{f_2 - f_1}{2f_1} t^2\right)\right\} \end{aligned} \quad (27)$$

f_1 が既知の場合は、この式が利用できる。しかし一般に任意の音では、 f_1 がわからない場合も多い。

そういった場合でも、元の音の m 倍まで変化させるという条件で変調したいという場合には、式 (27) を次のように書き換えればよい。(元の周波数を f_1 とする)

$$\sin\left\{2\pi f_1\left(t + \frac{m-1}{2} t^2\right)\right\} \quad (28)$$

正弦波の式を t の関数 $f(t)$ である、と見ると

$$f(t) = \sin(2\pi f t) \quad (29)$$

である。式 (28) は、この式の右辺の t が $t + \frac{m-1}{2} t^2$ に変わったものである。この変調を $g(t)$ であらわすと

$$g(t) = t + \frac{m-1}{2} t^2 \quad (30)$$

であり、 $f(t)$ を変調するというのは、 $f(g(t))$ と書ける。

ここで $m=2$ だとする。デジタルデータでは、 t は等間隔で変化する。サンプリング周波数が 8000Hz だとすると、0 秒からの最初の 3 点は $0, 1/8000, 2/8000$ となる。これらに対応する $g(t)$ は (13) (14) (15) となる。つまり、2 点目はもとの t に比べると (16) 秒分 (17) の値となっている。しかし、実際には、これが時刻 (14) での値となる。したがって、元の信号より少しずつ早く出力することになる。

自分で信号を 1 から作るような場合は、元の信号より少し早い値や遅い値もいくらかでも計算できる。しかし録音した音の場合は、任意の時刻の点を計算できない。このような場合は、実際にわかっている値でその周辺の値を推定するしかない。

そのようなときに利用できる方法に補間がある。最も簡単な補間は、近くの点を直線で結んでその直線上に必要な点があるとして計算する線型補間である。

例えば、 y という関数が $x=0$ のときに 3 $x=1$ のときに 5 という値であることがわかっているとす。このときに、 $x=0.5$ のときの値が知りたいとする。すると、 $(x, y) = (0, 3), (1, 5)$ を結ぶ直線の式は (18) である。その式を用いると $x=0.5$ のときの y の値は (19) となる。

MATLAB でこのような補間を計算する関数が `interp1` である。

```
>> x=[0 1];
>> y=[3 5];
>> x1=[0 0.5 1];
>> interp1(x,y,x1)
```

実行すると、 $x=0.5$ に対応する値がわかるはずである。

練習 65 下線部 (13) から (19) を埋めよ。

練習 66 `interp1` を利用して、自分で録音した音を最終的には、元の 2 倍 (1 オクターブ上) の音になるように変調せよ。

これと同じ考え方を利用して自分で録音した音にビブラートをかけることもできる。

a Hz を中心に、振幅 b 周波数 f_v の正弦波で上下するような波で周波数変調をおこなうときには周波数変調の関数 $f(t)$ は、

$$\begin{aligned}\frac{df(t)}{dt} &= a + b \sin(2\pi f_v t) \\ f(t) &= at + \underline{\hspace{2cm}} \quad (31)\end{aligned}$$

となる。

この $f(t)$ を式 (30) と同様に变形すると、

$$f(t) = a(t + \underline{\hspace{2cm}}) \quad (32)$$

となる。

任意の信号の場合、 a はわからない場合が多いが、 b を a の $1/10$ などと決めれば上記の例と同様に計算することができる。(上記の $\underline{\hspace{2cm}}$ から a, b を消去できる。)

そうなれば、こちらの場合も `interp1` を利用して周波数変調された信号を生成できる。

練習 67 下線部 (20), (21) を埋めよ。

練習 68 `interp1` を利用して、自分で録音した音にビブラートをかけよ。

20 ステレオ信号の操作

CD などのステレオ音源の RIFF 形式 (`.wav` という拡張子のついた形式) のファイルがあるとする。この音源にメインボーカルが入ったものだとすると、メインボーカルは大抵、中央に定位している (ヘッドホンで聞くと頭の真ん中で聞こえる)。中央に定位する音を c とする。この音は右チャンネルにも左チャンネルにも同じように含まれるため中央に聞こえる。

右チャンネルの音声波形は、右チャンネルの演奏の信号を r とおくと時間波形の重ね合わせなので、 $r + c$ であらわされる。同様に左チャンネルの信号は $l + c$ とあらわされる。

ここで、左チャンネルの信号から右チャンネルの信号を引く。すると、 $(l + c) - (r + c) = \underline{\hspace{2cm}}$ (22) となる。

この音はどのような音になるか、実際に聞いてみる。ステレオ音源を読み込んで、データが `data` サンプル周波数が `fs` に設定されているとする。(CD の場合は、`fs` は 44100)

ソースコード 21: 簡単カラオケ

```
>> ll=data(:,1);
>> rr=data(:,2);
>> kk=ll-rr;
>> soundsc(kk,fs)
```

練習 69 下線部 (22) を埋めよ。

練習 70 ソースコード 21 を実行してどのようになったか聴取せよ。

逆にメインボーカルだけを抜くことはできるだろうか? ソースコード 21 のような時間波形の簡単な演算では、なかなか c だけを残すことはできない。

楽曲によっては、オリジナルカラオケが手に入る場合もある。オリジナルカラオケが、左チャンネルが l 、右チャンネルが r に近いものなのであれば、簡単な時間波形の演算で c を得ることができる。

練習 71 オリジナルカラオケが手に入る楽曲があれば、それを使ってメインボーカルを抽出してみよ。その結果を実際に聴取して考察せよ。

ソースコード 21 で求めた信号 kk は、カラオケに聞こえる。しかし実際には、左チャンネルの演奏と右チャンネルの演奏を足したものではない。時間波形では、実際の演奏の波形ではないが、音としてはそれなりに聞こえる。したがって、スペクトルは、カラオケに近いものであると推定される。左チャンネルの信号から右チャンネルの信号を引いた信号の振幅スペクトルを K とする。

これを左チャンネルの振幅スペクトルから引いたものを C とする。これを時間領域に戻すことで、メインボーカルの信号とする。

フレームごとの処理は例えば、次のようになる。

ソースコード 22: 元信号の位相情報を用いた `ifft` による時間波形の再合成

```
1 >> K=fft(l-r); % l,r はステレオ信号の左右チャンネルに窓をかけたもの
2 >> phase=angle(K); % 位相を求める
3 >> K=abs(K); % 振幅スペクトルにする
4 >> L=abs(l); % 左チャンネルの振幅スペクトルを求める
5 >> C=L-K; % 左右に共通して含まれる信号を推定
6 >> c=real(ifft(C.*exp(1j*phase)));
```

6 行目の `real` は本来は不要である。

振幅スペクトルなど位相情報を無視して処理をおこなった場合に周波数領域から時間領域に信号を戻す場合には、位相情報が必要となる。このような場合に、処理前の位相が転用されることがある。

練習 72 ソースコード 22 を完成させよ。

練習 73 ソースコード 22 の考え方を利用してステレオ音源から中央に定位する成分を抜き出すプログラムを作成せよ。作成したプログラムの出力を聴取して、ボーカルだけを抜き出すためには他にどのような処理が必要かを考察せよ。

練習 74 練習 73 の考察結果を元に中央のボーカルを抜き出すプログラムを作成せよ。

21 エコーとリバーブ

音楽制作で利用されるエコーという効果は、ある一定の時間遅らせた出力を振幅を少し減少させて加えるものである。

この処理は、遅延と定数倍と加算で表現できる。つまり、デジタルフィルタとして実現できることがわかる。

例えば、BPM=120 の曲で 16 分音符一つつまり (23) ms 遅れて $-10/11$ 倍になってゆくフィルタを IIR フィルタとして実現する場合の漸化式は次のようになる。

$$y[n] = x[n] + \underline{\hspace{2cm}} \quad (24) \quad (33)$$

練習 75 下線部 (23), (24) を埋めよ。

練習 76 式 (33) にもとづいて、適当な音にエコーをかけてみよ。減衰の仕方や遅延時間をいろいろ変化させるとどうなるか試してみよ。

リバーブも似たようなしくみで実現できる。

リバーブは、ホールや部屋などの空間で、壁や床や天井などから反射されてくる音の重ねあわせの様子を模したものである。

この特性を示すインパルス応答は、特性を計測したい部屋で打撃音を録音することで調べることができる。

例えば、<http://www.slp.k.hosei.ac.jp/~itou/lecture/2007/ProjectA/impulse/stair-2-L.wav> は階段で収録した打撃音である。

この時間波形をサンプリング周波数をあわせた上で、FIR フィルタの係数として用いるとリバーブを模擬することができる。

練習 77 上記の打撃音と `filter` 関数を用いて、適当な音にリバーブをかけよ。サンプリング周波数をあわせるには `resample` 関数が役に立つ。

練習 78 上記の打撃音の波形に似た信号は、指数関数や乱数を組み合わせて作成できる。リバーブの効果を持つインパルス応答をいろいろと試作し、その効果を確認せよ。

課題

19、20、21 章で説明した手法、もしくは別の手法を用いてプログラムを作成せよ。

作成したプログラムについてのレポートを作成すること。また、そのレポートを説明するパワーポイントの資料を作成すること。

レポートには以下の項目を含むこと。

1. プログラムの説明
 - 目的
 - 原理 (工夫した部分についてソースコードを交えて説明)
2. 使用例
3. 考察

両方の資料を授業支援システムのみ Navi の第 8 回目の「準備課題」のところに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 11/13(土) 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・加工した場合には、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。