

サウンドデータの 周波数分析

本章ではサウンドデータの周波数分析について説明します。第1章では波形をそのまま観察するだけでしたが、離散フーリエ変換を利用すると、周波数という別の視点からサウンドデータの性質を調べることができるようになります。本章では離散フーリエ変換の原理について説明した後、MATLABを使ってサウンドデータの周波数分析を実際に体験してみます。

2.1 離散フーリエ変換

離散フーリエ変換は、本来アナログ信号を対象としていたフーリエ変換に、条件をつけてデジタル信号に適用したものです。フーリエ変換は - から + の時刻まで観測したアナログ信号の周波数特性を求めるための数学的方法で、その逆変換とともに次のように定義されています。

$$X(f) = \int_{-\infty}^{\infty} x(t) \exp[-j2\pi ft] dt \quad (-\infty \leq f \leq \infty) \dots\dots\dots (2.1)$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \exp[j2\pi ft] df \quad (-\infty \leq t \leq \infty) \dots\dots\dots (2.2)$$

ここで、 $x(t)$ は時間 t を変数とするアナログ信号、 $X(f)$ は周波数 f を変数とする $x(t)$ の周波数特性を表しています。式(2.1)はアナログ信号 $x(t)$ からその周波数特性 $X(f)$ を計算する「フーリエ変換 (Fourier Transform)」の定義式で、反対に式(2.2)は周波数特性 $X(f)$ からアナログ信号 $x(t)$ を計算する「逆フーリエ変換 (Inverse Fourier Transform)」の定義式となっています。

式(2.1)と式(2.2)をデジタル信号に適用することを考えてみましょう。デジタル信号は離散的にとびとびの時刻でしか値をもたないので、 t_s を標本化周期、 n を整数とすると、 t を nt_s に置き換えることができます。その結果、式(2.1)の積分記号をシグマ記号に書き換えることができます。また、 f_s を標本化周波数とすると、式(2.2)における f に関する積分範囲は $-f_s/2$ から $f_s/2$ となります。結果として式(2.1)と式(2.2)は次のようになります。

見本

$$X(f) = \sum_{n=-\infty}^{\infty} x(nt_s) \exp[-j2\pi fnt_s] \quad (-f_s/2 \leq f < f_s/2) \quad \dots\dots\dots (2.3)$$

$$x(nt_s) = \int_{-f_s/2}^{f_s/2} X(f) \exp[j2\pi fnt_s] df \quad (-\infty \leq n \leq \infty) \quad \dots\dots\dots (2.4)$$

これがデジタル信号のフーリエ変換とその逆変換の定義式になります。なお、デジタル信号では $X(f)$ は周期 f_s の周期信号になることが知られています。

ここで、 $x(n)$ が周期 N の周期信号であると仮定してみましょう。このとき、 $X(f)$ もまた周期 N の周期信号となり、標本化周波数 f_s を N で等分割した周波数にかぎり、値をもつことが知られています。すなわち、 k を整数とすると、 f を kf_s/N に置き換えることができます。また、第1章で述べたように、標本化周波数は標本化周期の逆数であることに注意すると、式(2.3)と式(2.4)は次のようになります。

$$X(kf_s/N) = \sum_{n=-N/2}^{N/2-1} x(nt_s) \exp\left[-\frac{j2\pi kn}{N}\right] \quad (-N/2 \leq k \leq N/2-1) \quad \dots\dots\dots (2.5)$$

$$x(nt_s) = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} X(kf_s/N) \exp\left[\frac{j2\pi kn}{N}\right] \quad (-N/2 \leq n \leq N/2-1) \quad \dots\dots\dots (2.6)$$

これが、「離散フーリエ変換 (DFT : Discrete Fourier Transform)」とその逆変換である「逆離散フーリエ変換 (IDFT : Inverse Discrete Fourier Transform)」の定義式です。

なお、DFT と IDFT の定義式は、シグマ記号のインデックスの範囲を0から $N-1$ に変更し、さらに、

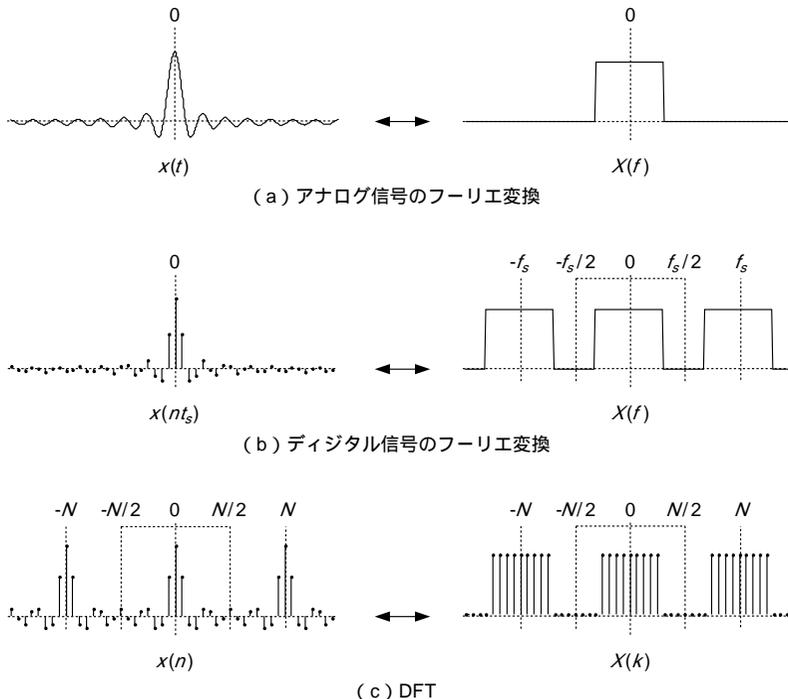


図2.1 フーリエ変換とDFT

見本

$x(nt_s)$ を $x(n)$, $X(kf_s/N)$ を $X(k)$ に書き換えた以下の式が一般的です .

$$X(k) = \sum_{n=0}^{N-1} x(n) \exp\left[-\frac{j2\pi kn}{N}\right] \quad (0 \leq k \leq N-1) \quad \dots\dots\dots (2.7)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \exp\left[\frac{j2\pi kn}{N}\right] \quad (0 \leq n \leq N-1) \quad \dots\dots\dots (2.8)$$

本書では式(2.7)と式(2.8)をDFTとIDFTの定義式として説明していくことにします .

図2.1にアナログ信号のフーリエ変換, デジタル信号のフーリエ変換, DFTについてまとめてみました . (a)のアナログ信号のフーリエ変換では, $x(t)$, $X(f)$ のどちらも連続信号になっていること, (b)のデジタル信号のフーリエ変換では, $x(nt_s)$ は離散信号ですが, $X(f)$ は周期 f_s の連続信号になっていること, (c)のDFTでは, $x(n)$, $X(k)$ のどちらも周期 N の離散信号になっていることが特徴となっており, 注意を払う必要があります .

2.2 振幅スペクトルと位相スペクトル

DFTを利用すると, 周波数という別の視点からサウンドデータの性質を調べることができるようになります . DFTによって計算される周波数特性が一体どのような意味をもっているのか, ここでは, DFTの定義式からその意味を考えてみることにしましょう .

DFTでは $x(n)$, $X(k)$ のどちらも複素数であるとしています . すなわち, 虚数単位を j とすると, $x(n)$, $X(k)$ はそれぞれ次のように定義されます .

$$x(n) = \text{real}[x(n)] + j \text{imag}[x(n)] \quad \dots\dots\dots (2.9)$$

$$X(k) = \text{real}[X(k)] + j \text{imag}[X(k)] \quad \dots\dots\dots (2.10)$$

ここで, 図2.2に示すように, $x(n)$, $X(k)$ を複素平面における極座標形式で表すと次のようになります .

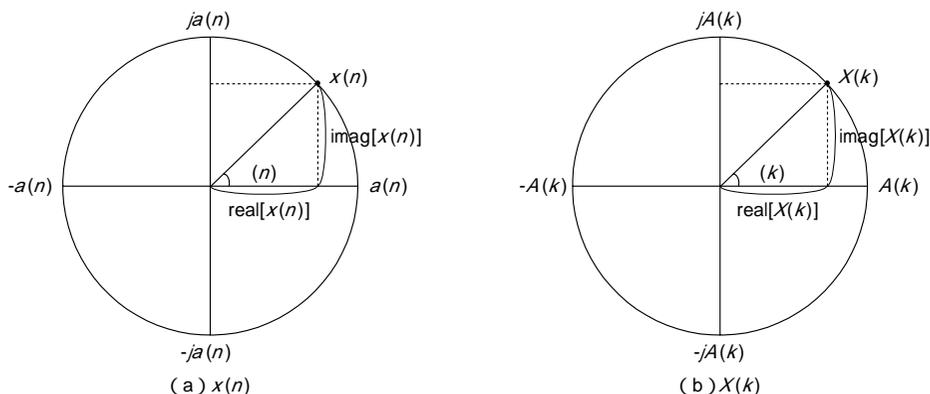


図2.2 複素数の極座標表現

$$x(n) = a(n) \exp[j\theta(n)] \dots\dots\dots (2.11)$$

$$X(k) = A(k) \exp[j\Theta(k)] \dots\dots\dots (2.12)$$

ただし，

$$\begin{cases} a(n) = \sqrt{\text{real}[x(n)]^2 + \text{imag}[x(n)]^2} \\ \theta(n) = \tan^{-1}[\text{imag}[x(n)] / \text{real}[x(n)]] \end{cases} \dots\dots\dots (2.13)$$

$$\begin{cases} A(k) = \sqrt{\text{real}[X(k)]^2 + \text{imag}[X(k)]^2} \\ \Theta(k) = \tan^{-1}[\text{imag}[X(k)] / \text{real}[X(k)]] \end{cases} \dots\dots\dots (2.14)$$

ここで， $a(n)$ と $\theta(n)$ は $x(n)$ の振幅と位相， $A(k)$ と $\Theta(k)$ は $X(k)$ の振幅と位相を表しています．

さて，式(2.12)を式(2.8)に代入すると次のようになります．

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} A(k) \exp\left[\frac{j2\pi kn}{N} + j\Theta(k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.15)$$

ここで，オイラーの公式を利用すると，式(2.15)は次のようになります．

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} A(k) \left\{ \cos\left[\frac{2\pi kn}{N} + \Theta(k)\right] + j \sin\left[\frac{2\pi kn}{N} + \Theta(k)\right] \right\} \quad (0 \leq n \leq N-1) \dots\dots\dots (2.16)$$

さらに， $x(n)$ をサウンドデータと仮定すると，サウンドデータは実数部のみをもつデジタル信号であることから，式(2.16)は次のようになります．

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} A(k) \cos\left[\frac{2\pi kn}{N} + \Theta(k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.17)$$

式(2.17)は，周波数 k/N の \cos 関数の重ね合わせによって $x(n)$ を表すことができることを意味しています．このとき， \cos 関数の振幅 $A(k)$ と位相 $\Theta(k)$ は DFT によって計算される周波数特性そのものとなっており，それぞれ「振幅スペクトル」，「位相スペクトル」と呼ばれます．

DFT では周波数の異なるいくつかの \cos 関数の重ね合わせによってサウンドデータを表すことができると仮定しています．DFT によって計算される周波数特性とは，そうした \cos 関数の振幅と位相の情報に他なりません．

2.3 離散フーリエ変換の計算

MATLAB を使って実際に DFT と IDFT をプログラミングしてみましょう．それぞれ，式(2.7)と式(2.8)をプログラミングすることになりますが，こうしたシグマ記号を使った計算は行列計算の形式に書き換えて考えると理解しやすくなります．

具体的に $N=8$ の場合について考えてみましょう． $N=8$ の場合について式(2.7)と式(2.8)を行列計算の形式に書き換えると，式(2.18)と式(2.19)になります．

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} \dots\dots\dots (2.18)$$

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix} = \frac{1}{8} \begin{bmatrix} W_8^0 & W_8^0 \\ W_8^0 & W_8^{-1} & W_8^{-2} & W_8^{-3} & W_8^{-4} & W_8^{-5} & W_8^{-6} & W_8^{-7} \\ W_8^0 & W_8^{-2} & W_8^{-4} & W_8^{-6} & W_8^{-8} & W_8^{-10} & W_8^{-12} & W_8^{-14} \\ W_8^0 & W_8^{-3} & W_8^{-6} & W_8^{-9} & W_8^{-12} & W_8^{-15} & W_8^{-18} & W_8^{-21} \\ W_8^0 & W_8^{-4} & W_8^{-8} & W_8^{-12} & W_8^{-16} & W_8^{-20} & W_8^{-24} & W_8^{-28} \\ W_8^0 & W_8^{-5} & W_8^{-10} & W_8^{-15} & W_8^{-20} & W_8^{-25} & W_8^{-30} & W_8^{-35} \\ W_8^0 & W_8^{-6} & W_8^{-12} & W_8^{-18} & W_8^{-24} & W_8^{-30} & W_8^{-36} & W_8^{-42} \\ W_8^0 & W_8^{-7} & W_8^{-14} & W_8^{-21} & W_8^{-28} & W_8^{-35} & W_8^{-42} & W_8^{-49} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} \dots\dots\dots (2.19)$$

ただし， W_8 は次のように定義されます．

$$W_8 = \exp\left[\frac{-j2\pi}{8}\right] \dots\dots\dots (2.20)$$

こうした行列計算は，for-end文を二重にすることでプログラミングできます．すなわち，MATLABでは，リスト2.1，リスト2.2のような関数MファイルによってDFTとIDFTを計算することができます．

実際にDFTの計算を行ってみましょう．そのための準備として，はじめにMファイルの実行環境を整えておきます．本書の付属CD-ROMのなかにあるchapter2フォルダを「マイコンピュータ」

リスト2.1 関数Mファイル：DFT_m

```

function [X_real,X_imag]=DFT_(x_real,x_imag,dft_size)
X_real=zeros(1,dft_size);
X_imag=zeros(1,dft_size);
for k=1:dft_size,
    for n=1:dft_size,
        w_real=cos(2*pi*(k-1)*(n-1)/dft_size);
        w_imag=-sin(2*pi*(k-1)*(n-1)/dft_size);
        X_real(k)=X_real(k)+w_real*x_real(n)-w_imag*x_imag(n);
        X_imag(k)=X_imag(k)+w_real*x_imag(n)+w_imag*x_real(n);
    end
end
    
```

リスト2.2 関数Mファイル：IDFT_.m

```
function [x_real,x_imag]=IDFT_(X_real,X_imag,dft_size)
x_real=zeros(1,dft_size);
x_imag=zeros(1,dft_size);
for n=1:dft_size,
    for k=1:dft_size,
        w_real=cos(2*pi*(k-1)*(n-1)/dft_size);
        w_imag=sin(2*pi*(k-1)*(n-1)/dft_size);
        x_real(n)=x_real(n)+w_real*X_real(k)-w_imag*X_imag(k);
        x_imag(n)=x_imag(n)+w_real*X_imag(k)+w_imag*X_real(k);
    end
    x_real(n)=x_real(n)/dft_size;
    x_imag(n)=x_imag(n)/dft_size;
end
```

リスト2.3 スクリプトMファイル：ex2_1.m

```
clear;
dft_size=60;
x_real=wavread('test.wav');
x_imag=zeros(1,dft_size);
[X_real,X_imag]=DFT_(x_real,x_imag,dft_size);
A=zeros(1,dft_size);
T=zeros(1,dft_size);
frequency=zeros(1,dft_size);
for k=1:dft_size,
    A(k)=sqrt(X_real(k)*X_real(k)+X_imag(k)*X_imag(k));
    T(k)=atan(X_imag(k)/X_real(k));
    frequency(k)=k-1;
end
subplot(2,1,1);
stem(frequency,A);
subplot(2,1,2);
stem(frequency,T);
```

「ローカルディスク (C:)」の中にコピーしてください。次に、第1章と同じ手順でchapter2フォルダに対するパスとカレントディレクトリの設定を行います。

準備ができたなら、chapter2フォルダにあるex2_1.m(リスト2.3)を開いてみましょう。Mファイルを開くには、ターミナルウィンドウの「ファイル」メニューから「開く」を選択します。

なお、DFT_.m、IDFT_.m、ex2_1.mには、いくつかのMATLABの関数が新しく登場しています。これらの関数を表2.1にまとめておきます。

それでは、ex2_1.m(リスト2.3)を実行してみましょう。このプログラムは、図2.3のサウンドデータ(test.wav)に対して $N=60$ のDFTを実行し、振幅スペクトルと位相スペクトルのグラフを表示します。実行結果を図2.4に示します。

図2.4をよく観察すると、 $k=N/2$ 、すなわち $k=30$ を中心として振幅スペクトルは線対称、位相スペクトルは点对称になっていることがわかります。 $k=N/2$ は $f_s/2$ 、すなわち標準化周波数の1/2となりま

表2.1 DFT_ .m , IDFT_ .m , ex2_1.m で新しく登場したMATLABの関数

関数	内容	文法
cos	余弦関数	cos_of_x=cos(x);
sin	正弦関数	sin_of_x=sin(x);
sqrt	平方根の計算	sqrt_of_x=sqrt(x);
atan	逆正接関数	atan_of_x=atan(x); % 戻り値は[- /2, /2]の範囲
subplot	複数のグラフの表示	subplot(M,N,p); % M行N列の複数のグラフを作成 % pによって描画するグラフを指定

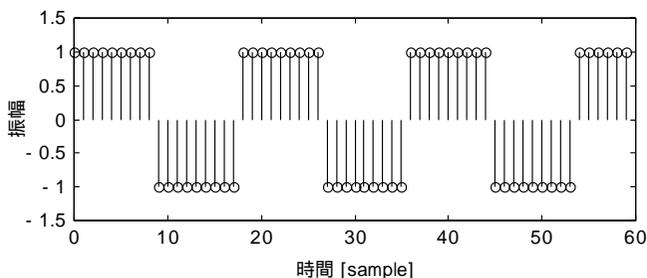


図2.3 テスト信号

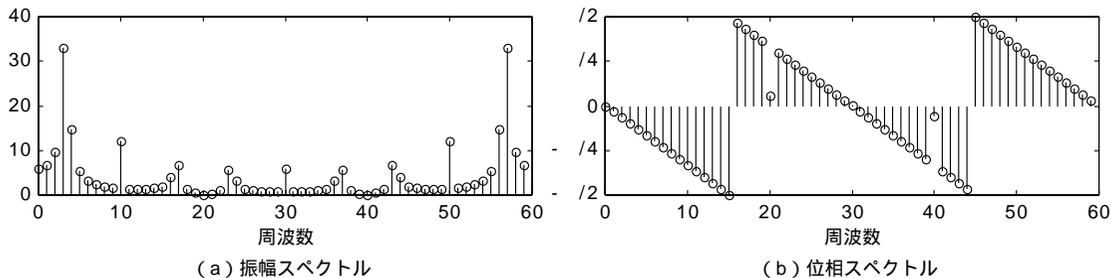


図2.4 テスト信号の周波数特性

すが、このようにDFTでは標準化周波数の1/2を中心として周波数成分が折り返すように現れることが特徴となっており、注意を払う必要があります。

第1章で述べたように、サウンドデータは標準化の際、エイリアス歪みを防ぐため、標準化周波数の1/2よりも大きい周波数成分を取り除いています。したがって、DFTの計算によって図2.4のように標準化周波数の1/2よりも大きい周波数成分が現れてしまうことを不思議に思われるかもしれませんが、たとえば、次のように標準化周波数の1/2以下の周波数成分だけで定義される信号を考えてみます。

$$x(n) = \sum_{k=0}^{N/2} A_0(k) \cos\left[\frac{2\pi kn}{N} + \Theta_0(k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.21)$$

式(2.21)は次のように展開できます。

$$\begin{aligned}
 x(n) = & A_0(0) \cos[\Theta_0(0)] + \frac{1}{2} \sum_{k=1}^{N/2-1} A_0(k) \cos\left[\frac{2\pi kn}{N} + \Theta_0(k)\right] + A_0(N/2) \cos[\pi n + \Theta_0(N/2)] \\
 & + \frac{1}{2} \sum_{k=N/2+1}^{N-1} A_0(N-k) \cos\left[\frac{2\pi kn}{N} - \Theta_0(N-k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.22)
 \end{aligned}$$

ここで、次のように定義します。

$$A_1(k) = \begin{cases} A_0(0) & (k=0) \\ \frac{1}{2} A_0(k) & (1 \leq k \leq N/2-1) \\ A_0(N/2) & (k=N/2) \\ \frac{1}{2} A_0(N-k) & (N/2+1 \leq k \leq N-1) \end{cases} \dots\dots\dots (2.23)$$

$$\Theta_1(k) = \begin{cases} \Theta_0(0) & (k=0) \\ \Theta_0(k) & (1 \leq k \leq N/2-1) \\ \Theta_0(N/2) & (k=N/2) \\ -\Theta_0(N-k) & (N/2+1 \leq k \leq N-1) \end{cases} \dots\dots\dots (2.24)$$

式(2.23)と式(2.24)を式(2.22)に代入すると、次のようになります。

$$\begin{aligned}
 x(n) = & A_1(0) \cos[\Theta_1(0)] + \sum_{k=1}^{N/2-1} A_1(k) \cos\left[\frac{2\pi kn}{N} + \Theta_1(k)\right] + A_1(N/2) \cos[\pi n + \Theta_1(N/2)] \\
 & + \sum_{k=N/2+1}^{N-1} A_1(k) \cos\left[\frac{2\pi kn}{N} + \Theta_1(k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.25)
 \end{aligned}$$

したがって、式(2.22)は次のように書き換えることができます。

$$x(n) = \sum_{k=0}^{N-1} A_1(k) \cos\left[\frac{2\pi kn}{N} + \Theta_1(k)\right] \quad (0 \leq n \leq N-1) \dots\dots\dots (2.26)$$

結果として、標準化周波数の1/2以下の周波数成分だけで定義される信号であっても、それよりも大きい周波数成分を含んだ信号と等しくなることがわかります。このとき、式(2.23)と式(2.24)に定義されているように、 $k=N/2$ を中心として振幅スペクトルは線対称、位相スペクトルは点对称になります。

このように、DFTでは標準化周波数の1/2以下の周波数成分だけで定義される信号であっても、それよりも大きい周波数成分が折り返すように現れることが特徴となっています。したがって、DFTによって計算された周波数特性からサウンドデータ本来の周波数特性を調べるには、標準化周波数の1/2以下の周波数成分だけに注目すればよいことに注意を払う必要があります。これは、DFTによって計算された周波数特性を解釈するうえで最も重要なポイントです。

2.4 サウンドデータの周波数特性

サウンドデータの周波数特性について、ここではとくに振幅スペクトルの特徴を調べてみることにしましょう。図2.5は、標準化周波数8kHz、量子化精度16bitで録音したギターのA音で、音の高さが220Hzとなっています。ex2_2.m(リスト2.4)は、このサウンドデータに対して $N=1024$ のDFTを突