

# デジタル信号処理(1)

035760A:横田敏明

平成 16 年 12 月 10 日

実験実施日：2004/12/3

報告書提出日：2004/12/10

## 1 目的

例題プログラムを実行することにより、Scilab の使いかたを習得するし、簡単な Scilab プログラムを作成することにより、デジタル信号処理の基礎を理解することを目的とする。

## 2 概要

アナログ信号をデジタル信号に変換するとき、時間によって刻々と変化を続ける物理量を高精度に表現できれば、質の高いデータといえる。すなわち、時間精度、物理量の精度である。音の場合、ボリューム (厳密には空気の圧力) をできるだけ細かく表現し、サンプリングレートをできるだけ短い時間に設定すれば、高い精度での標本化ができていないはずである。標本化定理については後述するが、信号波形の最短の周波数の半分以下のサンプリング周波数で十分であるといわれている。今回の実験は、Scilab の使用法を習得することがメインである。

## 3 報告事項

### 3.1 例題プログラムの解読・実行

```
f=1;           % 周波数
t=0:1/360:1;   % 0 から 1 までを 360 等分して代入していく
x=sin(2*pi*f*t); % 正弦波を作成
plot(t,x,'Time(sec)','Amplitude(v)') % 正弦波をプロット
Ts=0:1/100:1;  % 0 から 1 までを 100 等分して代入
xs=sin(2*pi*f*Ts); % 正弦波を作成
plot(Ts,xs);   % 正弦波をプロット

t=0:1/100:2*pi; % 0 から 2π までを 100 等分して代入
x=5*sin(t);     % 正弦波を x に代入
plot(t/2/pi,x); % 正弦波をプロット
% 適当な値を nn に代入
plot(t(1:19:nn)/2/pi,round(x(1:19:nn))); % 量子化してプロット
\\ stem(t(1:19:nn)/2/pi,round(x(1:19:nn)));
grid on;
```

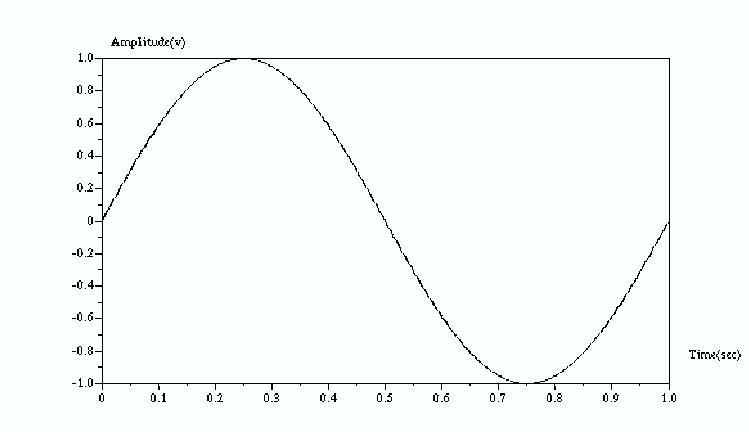


図 1: 例題 1 のプログラム

### 3.2 デジタル信号変換

以下のプログラムを実行した。

```
f=1;
w=2*%pi;
t=0:1/360:3;
x=cos(w*t)+2*cos(2*w*t)+cos(3*w*t)+2*sin(w*t);
plot(t,x,'Time(sec)', 'Amplitude(v)');
Ts=0:1/10:3;
xs=cos(w*Ts)+2*cos(2*w*Ts)+cos(3*w*Ts)+2*sin(w*Ts);
plot2d2(Ts,xs);
```

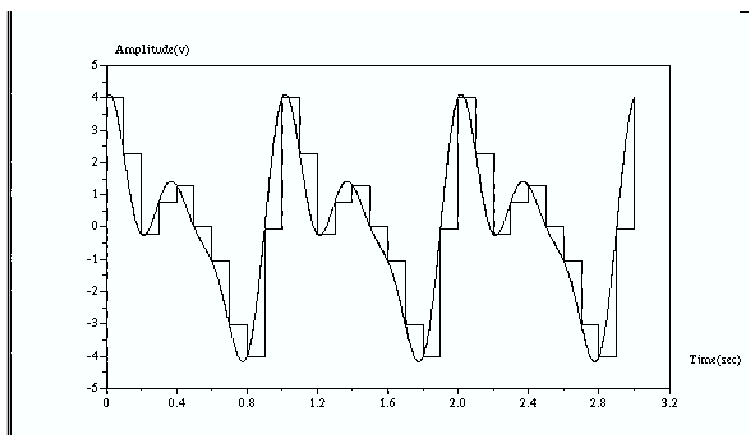


図 2:  $f_1(t)$  のグラフ

```

f=1;
w=2*%pi;
t=0:1/360:3;
x=1+3*cos(w*t)+2*cos(2*w*t)+cos(3*w*t)+2*sin(w*t);
plot(t,x,'Time(sec)', 'Amplitude(v)');
Ts=0:1/10:3;
xs=1+3*cos(w*Ts)+2*cos(2*w*Ts)+cos(3*w*Ts)+2*sin(w*Ts);
plot2d2(Ts,xs);

```

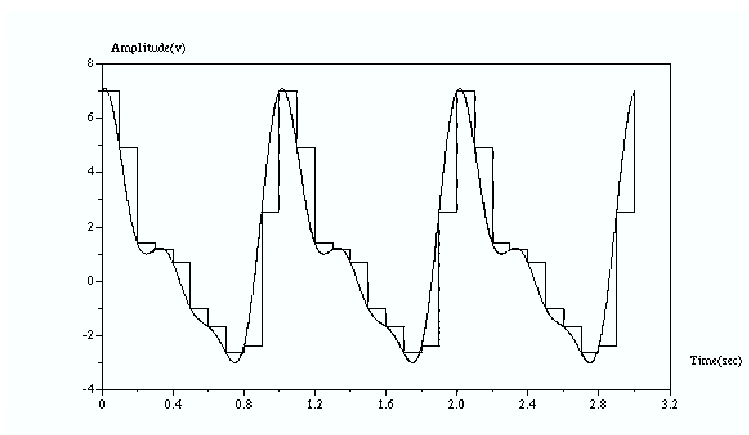


図 3:  $f_1(t)$  のグラフ

### 3.3 4bit 量子化

$f_1(t)$  に対して 4 ビットの量子化を行った.

```

f=1;
w=2*%pi;
t=0:1/360:3;
x=1+3*cos(w*t)+2*cos(2*w*t)+cos(3*w*t)+2*sin(w*t);
plot(t(1:16:100),round(x(1:16:100),'Time(sec)', 'Amplitude(v)'));
Ts=0:1/10:3;
xs=1+3*cos(w*Ts)+2*cos(2*w*Ts)+cos(3*w*Ts)+2*sin(w*Ts);
plot2d2(Ts,xs);

```

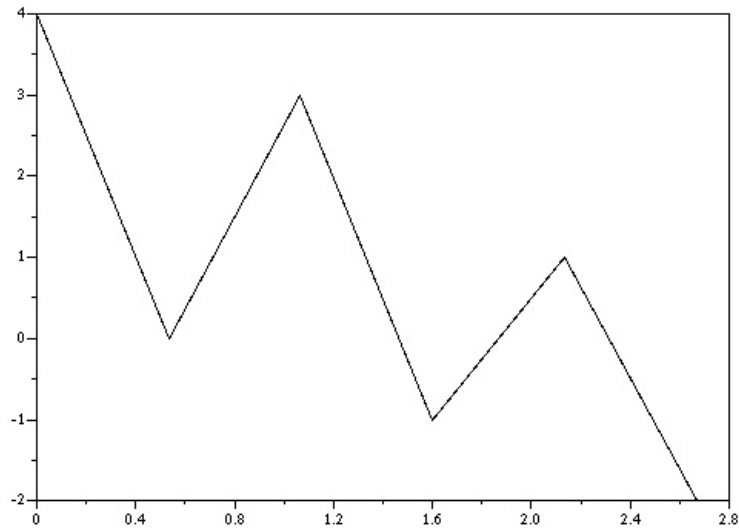


図 4: 量子化後のグラフ

### 3.4 信号の作成

$f_1(t) = \cos \omega_1 t$  と,  $f_2(t) = \cos \omega_2 t$  のデジタル信号を, 1[sec] 生成し, プロットした.  $\omega_1 = 1k[Hz]$ ,  $\omega_2 = 10k[Hz]$  とし, サンプルング間隔は 0.01m[sec] とする.

```
f=1000;    % 周波数 1kHz
w=2*pi;
t=0:1/100000:1;
x=cos(w*t*f);
plot(t,x);
```

のようにプログラムする. プロットしたグラフを見たときは「間違えたか?」と思った. しかし, 周波数が 1k と言うことは, 一秒間に 1000 往復上下している. よって, グラフが真黒になるのは当然である.

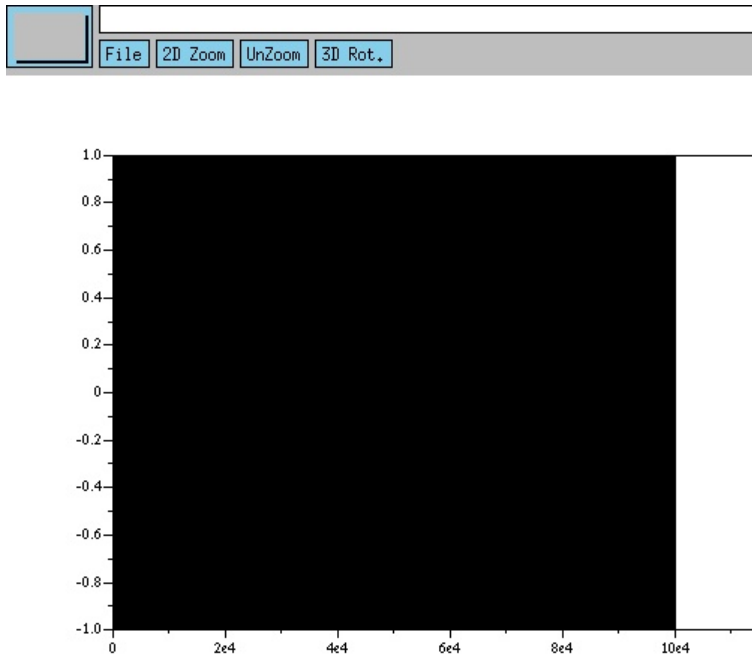


図 5:  $f_1(t)$  のグラフ

同じく,  $f_2(t)$  もグラフが真黒になるので省略する.

```
f=10000; % 周波数 10kHz
w=2*%pi;
t=0:1/100000:1;
x=cos(w*t*f);
plot(t,x);
```

## 4 標本化定理

あるアナログ的な信号をデジタルデータにサンプリングする場合、原信号に含まれる周波数成分をすべて正確にサンプリングするためには、原周波数の 2 倍以上のサンプリング周波数が必要となる。これを標本化定理という。たとえば、100Hz のアナログ信号を正確にサンプリングするためには、最低でも 200Hz 以上の周期でサンプリングしなければならない。もしサンプリング周波数の 1/2 よりも高い周波数成分が含まれていると、その成分がサンプリング結果にはエイリアス (alias、虚像) 信号として現われる。

## 5 パワースペクトル

ある音に、どのような周波数の純音が、それぞれどのくらいの強さで存在するかをグラフ化したものを、パワースペクトルと呼ぶ。カーステレオなどで表示されるグラフィックアナライザーは、パワースペクトルを数段の強弱表示で表したものだといえる。つまり、ある瞬間の音に含まれる周波数とボリューム(パワー)を解析するスペクトルである。

## 6 参考

- アスキー用語辞典
- <http://yougo.ascii24.com/>