

情報工学実験 2
デジタル信号処理
035743A : 比嘉雅樹

実験日 : 2004/12/06
提出日 : 2004/12/13

1 実験の目的

例題プログラミングを実行する事により、Scilab の使い方を習得する。また、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事を目的とする。

2 報告事項

2.1 配布資料の付録にある例題プログラムを読解し、実行せよ。

- 例題プログラム

```
f1=1; f2=20;  
t=[0:1/360:1];  
x1=cos(2*%pi*f1*t);  
x2=0.5*sin(2*%pi*f2*t);  
x3=x1+x2;  
subplot(2,2,1); plot2d3(t,x1);  
subplot(2,2,2); plot2d3(t,x2);  
subplot(2,2,3); plot2d3(t,x3);
```

このプログラムでは、画面を4つに分割し、 $\cos\omega t$ 、 $0.5\sin\omega t$ 、そして二つの波形を合わせたものを表示している。

- 実行結果

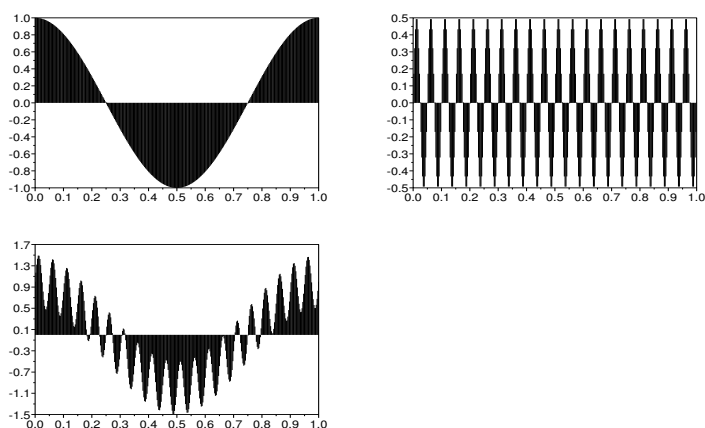


図 1: 例題プログラムの実行結果

2.2 $f_1(t) = \cos\omega t + 2\cos 2\omega t + \cos 3\omega t + 2\sin\omega t$ と $f_2(t) = 1 + 3\cos\omega t + 2\cos 2\omega t + \cos 3\omega t + 2\sin\omega t$ のアナログ信号を、各々、3 [sec] 間生成し、デジタル信号に変換せよ。また、プロットし比較せよ。ただし、基本周波数は1 [Hz] であり、サンプリング間隔 T は、0.1[sec] とする。

- $f_1(t)$ のデジタル信号変換プログラム

```
f = 1;
t = 0:1/10:3;
Ts = 0:1/360:3;
w = 2 * %pi * f;
x = cos(w*Ts) + 2*cos(2*w*Ts) + cos(3*w*Ts) + 2*sin(w*Ts);
subplot(1,2,1); plot2d(Ts,x);
F1 = cos(w*t) + 2*cos(2*w*t) + cos(3*w*t) + 2*sin(w*t);
subplot(1,2,2); plot2d3(t, F1);
```

- 実行結果

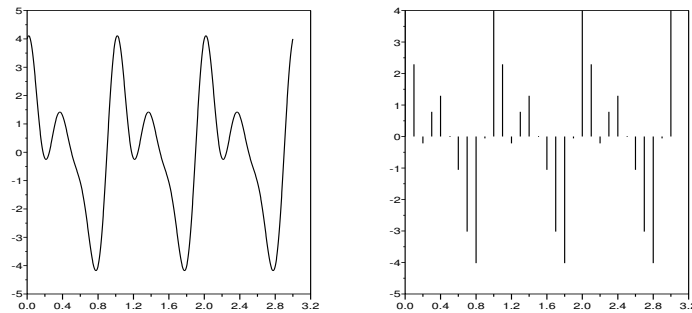


図 2: $f_1(t)$ のアナログ信号 (左) とデジタル信号 (右)

- $f_2(t)$ のデジタル信号変換プログラム

```
f = 1;
t = 0:1/360:3;
w = 2 * %pi * f;
Ts = 0:1/10:3;
x = 1 + 3*cos(w*t) + 2*cos(2*w*t) + cos(3*w*t) + 2*sin(w*t);
subplot(1,2,1); plot2d(t, x);
F2=1+ 3*cos(w*Ts) + 2*cos(2*w*Ts) + cos(3*w*Ts) + 2*sin(w*Ts);
subplot(1,2,2); plot2d3(Ts,F2);
```

- 実行結果

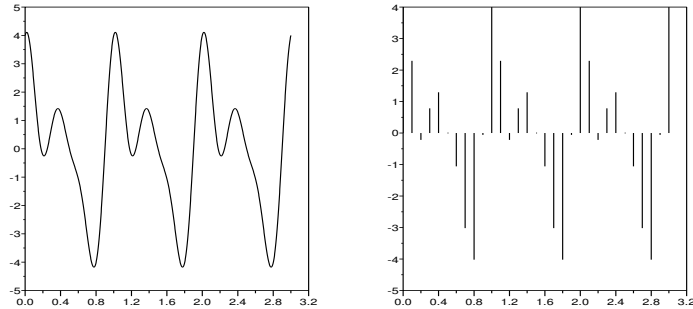


図 3: $f_2(t)$ のアナログ信号 (左) とデジタル信号 (右)

2.3 報告事項 2 の $f_1(t)$ で生成した信号に対して、4bit の量子化を行い、プロットせよ。

- プログラム

```
f = 1;
t = 0:1/100:3;
Ts=0:1/360:3;
w = 2 * %pi * f;
F1 = cos(w*t) + 2*cos(2*w*t) + cos(3*w*t) + 2*sin(w*t);
x = cos(w*Ts) + 2*cos(2*w*Ts) + cos(3*w*Ts) + 2*sin(w*Ts);
nn = length(t);
subplot(1,2,1); plot2d(Ts,x);
subplot(1,2,2); plot2d(t(1:16:nn), round(F1(1:16:nn)));
```

- 実行結果

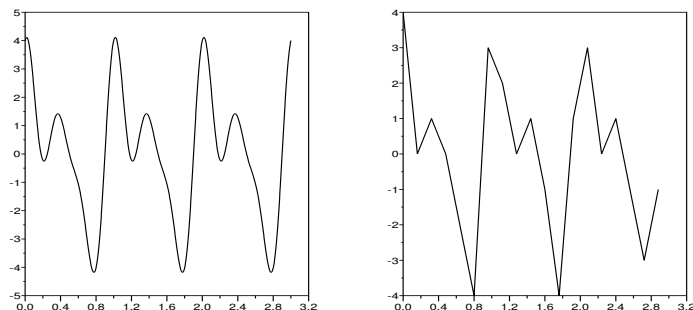


図 4: $f_1(t)$ のアナログ信号 (左) と量子化後 (右)

上の実行結果を見てみると、元の波形と量子化した波形は完全には一致しない事が分かる。それは、round を用いて値を近似しているためである。

2.4 $f_1(t) = \cos\omega_1 t$ と、 $f_2(t) = \cos\omega_2 t$ のデジタル信号を、各々1[sec]間生成し、プロットせよ。ただし、基本周波数 w_1, w_2 は、各々1k[Hz],10k[Hz] であり、サンプリング間隔 T は、 $1/44100$ [sec] とする。

- $f_1(t)$ 表示プログラム

```
f1 = 1000;
t = 0:1/44100:1;
w1 = 2*%pi*f1;
F1 = cos(w1*t);
plot2d(t,F1);
```

- 実行結果

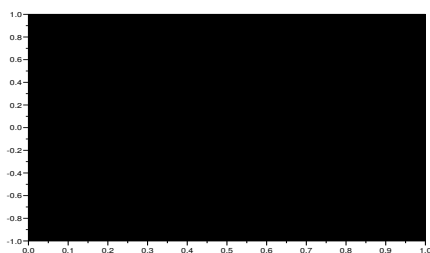


図 5: プログラムの実行結果

- $f_2(t)$ 表示プログラム

```
f2 = 10000;
t = 0:1/44100:1;
w2 = 2*%pi*f2;
F2 = cos(w2*t);
plot2d(t,F2);
```

- 実行結果

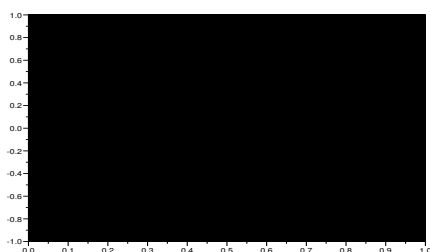


図 6: プログラムの実行結果

2つのプログラムの実行結果が真っ黒になるのは、周波数が1kHzと大きく、またサンプリング間隔も $1/44100$ sと小さいからである。ここで、サンプリング間隔を $1/44100$ [sec]としたのは、次の課題でwavに変換した時に音が聞き取れる様にするためである。

2.5 報告事故4で生成した信号($f_1(t)$, $f_2(t)$)を、各々wavに変換し、聞いてみる事。

- プログラム

```
f1 = 1000; f2 = 10000;  
t = [0:1/44100:1];  
w1 = 2*%pi*f1;  
w2 = 2*%pi*f2;  
F1 = cos(w1*t);  
savewave('F1.wav',F1);  
F2 = cos(w2*t);  
savewave('F2.wav',F2);
```

上記のプログラムを実行し、wavに変換して聞いてみた。音は $f_2(t)$ の方が $f_1(t)$ よりも高い音が聞こえた。

2.6 標本化定理 (sampling theorem) について調べよ。

入力信号の性質に応じてどのような周波数で標本化すればよいか、その際に標本化定理を満たさなければならない。標本化定理とは、「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、元の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、入力信号に含まれる最高の周波数成分の2倍以上のサンプリング周波数が必要となる。」という定理で、デジタル信号処理ではとても重要である。また、この定理に違反すると、エイリアシング (Aliasing) と呼ばれる現象が生じる。図7にエイリアシングの図を示す。

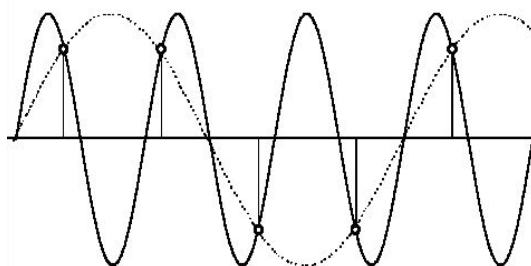


図 7: エイリアシングの図

エイリアシングとは、上の図の実線の信号を、それより低い周波数で標本化すると、あたかも点線のような低い周波数成分が存在するかのように見えてしまう現象である。データサンプリングの際のエイリアシングは、サンプリングの前に元の信号に含まれるサンプリング周波数の1/2以上の周波数成分をあらかじめ除去しておくことで避ける事ができる。

2.7 パワースペクトル (power spectral) について調べよ。

ある音に、どのような周波数の音が、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いう。ある波形を完全に表現する時にはパワースペクトルと位相スペクトルの両方が必要である。位相スペクトルとは、それぞれの音の時間的なずれを表したもの。また、波動で表現できる物質すべてにこの概念は適用できる。

参考文献

- [1] <http://www.brl.ntt.co.jp/IllusionForum/basics/audisense/content02.html>
- [2] <http://laputa.cs.shinshu-u.ac.jp/~yizawa/InfSys1/basic/chap2/>
- [3] <http://yougo.ascii24.com/>
- [4] http://www.ok.net.it-chiba.ac.jp/okawa/sp/dsp08_samp_theorem.html