データ解析（国際開発）
(3) 時間関数の離散化（実習）

高田潤一
takada@ide.titech.ac.jp

2010 年 4 月 16 日

概 要

本日は，前回の説明が終わらなかった呼ばれる時間関数の離散化を説明し，PC により実習を行う．

1 標本化（補足・訂正）

1.1 標本化とスペクトル

連続的な時間関数を，適当な時間間隔で標本化 (sampling) したときに，元の時間関数を忠実に表現できる条件を考える．

1.1.1 インパルス列とその性質

インパルス列[1]

$$d(t) = T_{\mathrm{s}} \sum_{k=-\infty}^{\infty} \delta(t - kT_{\mathrm{s}}) \tag{1}$$

は周期 T_{s} の周期関数となるので，次のようにフーリエ級数展開可能である．

$$\begin{aligned} d(t) &= \sum_{n=-\infty}^{\infty} d_n \exp\left(\mathrm{i}2\pi n \frac{t}{T_{\mathrm{s}}}\right) \\ &= \sum_{n=-\infty}^{\infty} d_n \exp(\mathrm{i}2\pi n f_{\mathrm{s}} t) \qquad (2) \\ d_n &= \frac{1}{T_{\mathrm{s}}} \int_{-\frac{T_{\mathrm{s}}}{2}}^{\frac{T_{\mathrm{s}}}{2}} d(t) \exp\left(-\mathrm{i}2\pi n \frac{t}{T_{\mathrm{s}}}\right) \mathrm{d}t \\ &= \int_{-\frac{T_{\mathrm{s}}}{2}}^{\frac{T_{\mathrm{s}}}{2}} \delta(t) \exp\left(-\mathrm{i}2\pi n \frac{t}{T_{\mathrm{s}}}\right) \mathrm{d}t \\ &= 1 \qquad (3) \end{aligned}$$

[1]前回の定義とは T_{s} 倍だけ異なる．これは，

$$\lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} d(t)\mathrm{d}t = 1$$

となるよう正規化したためである．

すなわち，

$$d(t) = \sum_{n=-\infty}^{\infty} \exp(\mathrm{i}2\pi n f_s t) \tag{4}$$

である．ただし，$f_s = \frac{1}{T_s}$ を標本化周波数という．

これを用いると，インパルス列 $d(t)$ のフーリエ変換 $D(f)$ も次のようにインパルス列になる．

$$\begin{aligned} D(f) &= \int_{-\infty}^{\infty} d(t) \exp(-\mathrm{i}2\pi f t)\mathrm{d}t \\ &= \int_{-\infty}^{\infty} \left(\sum_{n=-\infty}^{\infty} \exp(\mathrm{i}2\pi n f_s t) \right) \exp(-\mathrm{i}2\pi f t)\mathrm{d}t \\ &= \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left[-\mathrm{i}2\pi (f - n f_s) t)\right] \mathrm{d}t \\ &= \sum_{n=-\infty}^{\infty} \delta(f - n f_s) \end{aligned} \tag{5}$$

なお，最後の等号は

$$\int_{-\infty}^{\infty} \delta(f - f_0) \exp(\mathrm{i}2\pi f t)\mathrm{d}f = \exp(\mathrm{i}2\pi f_0 t) \tag{6}$$

より自明である．

1.1.2 標本化した関数とスペクトル

連続関数 $x(t)$ を間隔 T_s で標本化した関数 $x_s(t)$ はインパルス列 $d(t)$ を用いて次の式で表される．

$$x_s(t) = x(t)d(t) = x(t) \sum_{k=-\infty}^{\infty} T_s \delta(t - kT_s) = T_s \sum_{k=-\infty}^{\infty} x(kT_s)\delta(t - kT_s) = T_s \sum_{k=-\infty}^{\infty} x_k \delta(t - kT_s) \tag{7}$$

ただし，$x_k = x(kT_s)$ は標本化された時系列である．

したがって，標本化した関数 $x_s(t)$ のスペクトル $X_s(f)$ は，次の式で表される．

$$\begin{aligned} X_s(f) &= X(f) * \sum_{n=-\infty}^{\infty} \delta(f - n f_s) \\ &= \int_{-\infty}^{\infty} X(\nu) \sum_{n=-\infty}^{\infty} \delta(f - \nu - n f_s)\mathrm{d}\nu \\ &= \sum_{n=-\infty}^{\infty} X(f + n f_s) \\ &= \sum_{n=-\infty}^{\infty} X(f - n f_s) \end{aligned} \tag{8}$$

すなわち，$X_s(f)$ は $X(f)$ を f_s 間隔で平行移動したものの重ね合わせで表される周期関数となる．

1.2 再構成とエリアジング

1.2.1 完全再構成

時間関数 $x(t)$ が周波数 $\frac{w}{2}$ で帯域制限されている，すなわち

$$X(f) = 0,\ f > \frac{w}{2} \tag{9}$$

と仮定する．

もしも $f_s < w$ であれば，スペクトル $X_s(f)$ は周波数軸上で分離し，

$$X_s(f) = X(f), \ -\frac{f_s}{2} < f < \frac{f_s}{2} \tag{10}$$

が成り立つ．すなわち，理想的な低域通過フィルタ

$$H(f) = \begin{cases} 1, & |f| < \frac{f_s}{2} \\ 0, & |f| > \frac{f_s}{2} \end{cases} \tag{11}$$

を用いることにより，

$$X(f) = X_s(f)H(f) \tag{12}$$

が成立する．すなわち，この場合は，標本化された時系列から元の時間関数を完全に再構成することが可能となる．低域通過フィルタのインパルス応答は次の式で表される．

$$\begin{aligned} h(t) &= \int_{-\infty}^{\infty} H(f)\exp(\mathrm{i}2\pi f t)\mathrm{d}f \\ &= \int_{-\frac{f_s}{2}}^{\frac{f_s}{2}} \exp(\mathrm{i}2\pi f t)\mathrm{d}f \\ &= \frac{\sin \pi f_s t}{\pi t} \\ &= f_s \mathrm{sinc}(f_s t) \end{aligned} \tag{13}$$

ただし，sinc 関数は次のように定義される．

$$\mathrm{sinc}x = \frac{\sin \pi x}{\pi x} \tag{14}$$

sinc 関数は引数が 0 のときに 1，それ以外の整数のときに 0 となる性質をもつ．元の時間関数 $x(t)$ は標本化した時間関数 $x_s(t)$ を用いて，次のように再構成される．

$$\begin{aligned} x(t) &= x_s(t) * h(t) \\ &= T_s \sum_{k=-\infty}^{\infty} x_k \delta(t - kT_s) * f_s \mathrm{sinc}(f_s t) \\ &= \sum_{k=-\infty}^{\infty} x_k \mathrm{sinc}\{f_s(t - kT_s)\} \end{aligned} \tag{15}$$

このように，標本化間隔 T_s を $\frac{1}{w}$ 以下で標本化した場合には，補間関数として sinc 関数を用いることにより，元の時間波形の完全再構成が可能であることを**染谷・シャノンの標本化定理**といい，標本間隔の上限 $\frac{1}{w}$ を**ナイキスト間隔**という．

1.2.2 エリアジング

一方， $f_s > w$ の場合には，スペクトル $X_s(f)$ は周波数軸上で分離せず，互いに重複したものとなってしまう．この場合に，sinc 関数を用いて補間を行うと，w 以上の周波数成分が，すべて f_s の整数倍だけ周波数シフトして重畳されたように見える．このような現象をエリアジングと呼ぶ．

2　実習：標本化

標本化，再構成，エリアジングを実体験するために，Scilab を使用した簡単な実習を行う．まず Scilab に関する簡単な説明を行い，続いて正弦波の標本化と再構成について説明する．

2.1　Scilab

Scilab は Matlab 同様に数値計算のために開発された統合環境ソフトウェアである．元々フランスの数学研究所である INRIA が開発してきたが，現在は，Scilab Sonsortium が主体となって開発を行っている．Windows の他，Linux 及び MacOSX でも動作する．

従来，高度な数値計算には FORTRAN や c といったプログラミング言語が使用されてきた．これらのプログラミング言語ではコンパイラが使用され，プログラムを実行可能な形式に変換する必要があり，変数のモニタリングなどにはデバッガを使用しなくてはならず，プログラミングの手間が大きい．また，汎用の計算機言語では，行列・ベクトル，特殊関数などは言語の仕様に含まれておらず，数値計算ライブラリを使用する必要があり，これらもコンパイル時にライブラリとしてリンクする必要がある．さらに，グラフィック機能も標準の言語仕様には含まれておらず，コンパイラとともに提供されているグラフィックライブラリを使用する必要があり，コンパイラ毎に独自の拡張となっている．

Scilab はインタプリタ言語であり，コマンドラインから命令を直接実行することも，プログラムを逐次的に動かすことも可能である．数値計算ソフトウェアである Scilab は行列計算や特殊関数を内蔵しており，ライブラリなしに使用することが可能である．特にグラフの描画機能を内蔵しており，計算結果をプログラム中で可視化することが容易である．

2.2　使用ガイド

英語版チュートリアルは http://www.scilab.org/content/download/1104/10840/file/introscilab.pdf からダウンロード可能である．最近は日本語による入門書も多数出版されている．

Scilab を起動すると，コンソール画面が表示される．簡単な計算はプログラムを組まなくてもコンソール画面からコマンドを入力することにより実行可能である．

ヘルプブラウザ　help コマンドを入力することにより，ヘルプブラウザが表示される．

メモリクリア　clear コマンドを入力することにより，過去に入力したデータが消去される．

コマンドラインの編集と履歴　コマンドラインは，通常の Windows ソフトウェアと同じく左右矢印キー，BS キー，DEL キーなどを使用して編集可能である．また，上下矢印キーを使用すれば，過去に実行したコマンドの履歴が表示され，これも編集して新たなコマンドとして利用できる．なお，UNIX/Linux でプログラマに広く使われている GNU Emacs と同じキーバインドも使用できる．

結果表示と省略　コマンドラインを実行すると実行結果が画面に表示される．邪魔な場合は，セミコロン “;” で終端することにより，結果が表示されなくなる．

変数　Scilab では型宣言は必要としない．

スカラ `a = 1`

行ベクトル `a = [1, 2, 3]`

列ベクトル `a = [2; 4; 6]`

行列 `a = [1, 2; 3, 4]`

特殊定数　Scilab で定義済の変数は “%” で始まる．代表的なものとしては，虚数単位 `%i` や円周率 `%pi` がある．

算術演算　和 “+”，差 “-”，積 “*”，商 “/”，べき乗 “^” が使用できる．スカラだけでなく，行列に対しても使用可能である．

組み込み関数　三角関数 “sin”, “cos”，指数関数 “exp” などはベクトルや行列にも適用可能であり，この場合は各要素に対して関数が適用される．

グラフの描画　グラフは “plot(a,b)” のように，横軸及び縦軸を表すベクトルの組を指定することにより描画できる．なお，グラフコマンドは重ね書きされるため，前のグラフを消去したい場合は “clf()” を使用する．

2.3　標本化

三角関数を描いてみよう．要素を等間隔に発生させるには，`t = [0:0.1*%pi:10*%pi]` のようにコロン “:” を使用し，[初期値:増分:終了値] のように入力する．

```
clf();
t = [0:0.1*%pi:20*pi];
x = cos(t);
plot(t,x);
```

上記の例では，周期 2π の余弦関数が 10 周期にわたって出力される．このとき増分が標本化間隔である．そこで，ナイキスト間隔が π であることを意識し，増分を 0.25π, 0.5π, 0.75π, π, 1.5π, 2π, 2.5π と変化させ，サンプリングの結果を考察せよ．

2.4　再構成

次に sinc 関数による再構成を行ってみよう．sinc 関数による補間の効果が判るように，標本化間隔の $\frac{1}{10}$ の間隔で再構成を行う．

```
clear;
clf();
ts = 0.1 * %pi;  // sampling interval
fs = 1 / ts;     // sampling frequency
```

```
td = 0.1 * ts;   // reconstruction
tmax = 10 * %pi; // upper bound
tr =[0:td:tmax]; // reconstructed time step
x = zeros(tr);   // initialization
for t = 0:ts:tmax do
  x = x + cos(t) * sinc(fs * (tr - t * ones(tr)) * %pi);
  // sinc(x)= sin(x) / x in scilab
end;
plot(tr,x);
plot(tr,cos(tr));
```

上記プログラムでは，周期 2π の余弦関数を sinc 関数により再構成している．ts を 0.25π, 0.5π, 0.75π, π, 1.5π, 2π, 2.5π と変化させて，再構成・エリアシングの様子を観測せよ．

課題

演習の結果をレポートとしてまとめよ．なお，講義時間内に終了しなかった場合は IDE ラウンジもしくは自宅で完成させること．

締切は 4 月 23 日（金）9:00 とし，講義開始前に回収する．前日までに提出する場合には，メールボックス S6-4（南 6 号館ロッカー室横）に提出のこと．表紙は必要ないが，大きさは A4 判とし，学籍番号・氏名・提出日を上部欄外に記入すること．

参考文献

[1] 斉藤洋一，信号とシステム，第 6 章，コロナ社，2003.

[2] 大野修一，Scilab 入門，CQ 出版社，2009.