ひたすら楽して
音響信号解析

— MATLABで学ぶ基礎理論と実装 —

博士（工学）　森勢　将雅　【著】

コ ロ ナ 社

ま　え　が　き

何十年も前から，偉大なる信号処理研究者たちの手によって，数多の書籍が出版されてきた。Web で検索すれば，信号解析や関連トピックを扱う膨大な数の書籍や Web の解説記事が生まれており，どの本が学びやすいか探すだけでも大変であろう。信号処理の書籍は，入門書，おもに洋書で出版される分厚い専門書，その他諸々に分けられる。本書は，「自分で読みたい教科書を作る」という筆者の情熱を出発点に，信号解析の基礎をひたすら楽して習得することを目指して工夫を凝らしている。

フーリエ変換を回避して高速フーリエ変換（FFT：fast Fourier transform）によるスペクトル解析を習得せよ，という目標は無謀に映るかもしれない。信号解析を習得するために必要な知識は多いため，解析したい信号があるにもかかわらず延々と基礎勉強を続けることは，苦行に感じることだろう。本書は，このような基礎勉強が苦行という懸念を覆し，プログラムにより実践的な内容に絞って習得することを目指している。逆に考えて，先人たちが実装したプログラムを利用すれば信号解析をするだけなら可能である，と割り切って執筆したことが特色である。

本書は，筆者の専門分野である 1 次元の音響信号解析に特化し，数式をプログラムで実装することに比重を置いた構成を採用した。これは，教科書に記載された数式をどのようにプログラムに落とし込めばよいのかわからずに躓く学生がこれまで多かった，という筆者自身の経験に基づく。数式と同じ結果が得られるよう理論を厳密に実装することは，アルゴリズムによっては容易ではない。そのため，例となるプログラムを数式に合わせて掲載することで，数式とプログラムの対応付けができることを目指している。信号解析の基礎を扱うという都合上，大学学部 2, 3 年生の講義で用いる教科書を想定している。

ひたすら楽するというコンセプトから，本書はオーソドックスなディジタル信号処理の教科書とは異なる道筋で説明を展開することもある。例えば，前述の高速フーリエ変換を習得するためには，背景にある理論として学ぶべきことが多数存在する。おそらく多くの教科書では，フーリエ級数展開を出発点に複素フーリエ級数展開へ進み，そこからフーリエ変換の解説に進む。その後，離散フーリエ変換（DFT：discrete Fourier transform）を解説してからようやく高速フーリエ変換の説明に至る。場合によっては，アナログ信号処理に関する基礎やラプラス変換，z 変換が入るかもしれない。一方本書は，例えば「高速フーリエ変換によるスペクトル解析」を初めにゴールとして定める。ゴールに必要な理論に限定して一切の無駄を省いた説明によりただひたすら楽をする，そんな教科書を意識した。楽をするため一般的な書籍に存在する章末問題すら省き，プログラムの書き写しにより基礎的な信号解析を習得できる構成にした。

もう 1 点の特色は，本書で扱うプログラミング言語に MATLAB を採用していることである。本書を執筆している 2020 年であれば，信号解析を学ぶための筆頭候補に挙がるプログラミング言語は Python であろう。Python は deep learning 分野でシェアを伸ばしているプログラミング言語で，信号解析に関するライブラリも豊富にリリースされている。MATLAB を選んだ理由は筆者の好みであるが，別の理由として，やはり楽をするための完成度の高さが挙げられる。MATLAB はフリーソフトではないが，FreeMat や Octave などの MATLAB クローンがフリーで利用できる。本書に記載した例題は，MATLAB 以外では実装されていない関数や乱数を用いる場合を除き，MATLAB (2020a) と FreeMat (v4.2) でほぼ同じ出力が得られることを確認している。信号解析の入門に用いる言語として，たぶん MATLAB が一番楽だと思います。

最後に，本書を執筆する機会を与えてくださった皆様に深謝します。また，相変わらず年中自由気ままな時間に執筆することを許してくれた妻に感謝します。

2020 年 12 月

森勢　将雅

目　　　次

3. 離散フーリエ変換の考え方

4. 高速フーリエ変換によるスペクトル解析

5. 窓　関　数

6. ディジタルフィルタ

7. 信号の種類に応じた解析法

1. 基礎知識

本書では，数式とプログラムを多数掲載する。プログラムは MATLAB で動作するものであり，例えば以下のように記載する。

```
>> x=1;
```

ここで，>>の後に記載されているプログラムは，読者がコマンドウィンドウに打ち込むことを想定している。本書を読み進めるにあたり，数式を解きながら併せて記載されたプログラムも順次実行することで，数式とプログラムとを対応付けて学べるように工夫した。基本的に各プログラムは単独で動作するようにしているが，記述を短縮するため一部関数化することもある。本書では，MATLAB に関する最低限の知識は備わっていることを前提にしているため，細かい仕様について解説はしない。また，いわゆる MATLAB 的な記載により短いプログラムで高速に動作させることも可能であるが，可読性と数式との対応を意識した実装を優先する。各セクションのタイトルから想像できる内容を理解できていると感じ，MATLAB の知識も十分にある読者は，楽をするため 1 章をスキップしても問題ない。

1.1　離散信号と波形の表示

現実世界に存在する音は**連続信号**（analog signal）であり，**離散信号**（digital signal）に変換後，計算機に取り込まれる。この操作を **A-D 変換**（Analog-to-Digital conversion や A-D conversion とも）と呼び，A と D は連続信号と離散信号の英語表記に由来する。A-D 変換の結果，計算機上では単なる数字の羅列として音を記録していることになる。例えば，記録された数値が，先頭から順に「1, 3, −5, 2」であったとしよう。これを MATLAB の配列に格納する場

合のプログラムは以下となる。

```
>> x=[1,3,-5,2];
```

ここで，各数字は計算機に記録された音響信号の振幅に相当する。末尾にセミコロンを付けているが，MATLAB ではセミコロンを外してもエラーとはならない。セミコロンを外した場合は，計算結果がコマンドウィンドウに表示される。おもにデバッグにおいて，計算結果を逐次確認する際に役立つ仕様である。

現実世界の時刻とプログラム上で格納する配列の添え字については注意が必要なため，本書のコンセプトである数式とプログラムを対応付ける観点からもう少し説明を加える。上述の例で，先頭の振幅 1 を記録した時間を考えることにする。「何月何日の何時何分何秒 $\cdots$」という時刻に記録したことは事実だとしても，計算機上でファイル保存する際に，すべての振幅にこのような時刻を記録することはまずない。多くの場合，絶対的な時刻を持たず振幅値のみが順番に保存されることになり，先頭の振幅を原点（0 秒）における値として扱う。このイメージは，**図 1.1** のようになる。

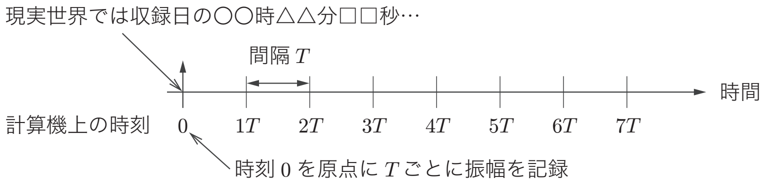


図 1.1　現実世界の時刻と計算機上の時刻との対応付け

今回の例では，振幅値 1 を記録した時刻が原点である。C 言語などのプログラミング言語では，配列の先頭は `x[0]` であり 0 番目としてアクセスする。一方，MATLAB における配列は先頭が 1 番目であるため，`x(1)` としてアクセスする必要がある。MATLAB を使う際には，この差がバグの原因にもなるので特に注意が必要である。なお本書では，上記のように，数式として扱う変数とプログラムとでは異なるフォントで記載する。

図 1.1 で示したように，A-D 変換では，振幅の記録を特定の時間間隔 T ごとに実施している。この時間間隔のことを**標本化周期**（sampling period）と呼

ぶ。この逆数，つまり 1 秒当たりの振幅値の数に相当する値が**標本化周波数**（sampling frequency でサンプリング周波数ともいう）である。振幅を 10 ms ごとに記録している場合，標本化周期は 10 ms，標本化周波数は 100 Hz となる。

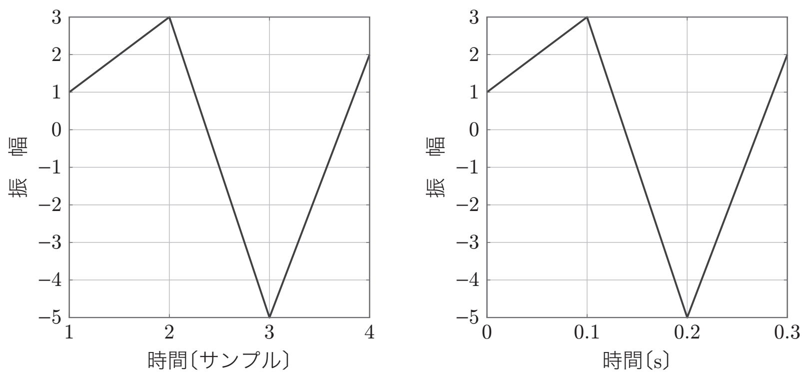
波形を表示する際は, 横軸の単位を秒などの時間にすることが望ましい。MATLAB で波形を表示する代表的な方法は **plot 関数**であり

```
>> plot(x);
```

と記載する。plot 関数では，引数が一つの場合，横軸を 1 から 4 サンプルと見なした数字で表示する。以下のプログラムでは，標本化周波数を 10 Hz に設定し，横軸を秒を単位とした時間軸として波形を表示する。

```
>> fs=10;
>> t=(0:3)/fs;
>> plot(t,x);
```

`fs` は標本化周波数に対応し，また，この変数名は慣習的に標本化周波数を記述するために用いられる。`t` は n 番目の振幅が何秒の値であるかを記録する配列である。この例では，振幅の先頭を原点として扱っており，先頭の振幅から時刻 0, 0.1, 0.2, 0.3 秒に対応する。**図 1.2** は，横軸をサンプルにした場合と，秒にした場合との表示の比較である。今回のプログラムでは省略しているが，横軸



（a）　横軸をサンプルにした場合　（b）　横軸を秒にした場合

図 1.2　横軸の単位をサンプルと秒にして表示した 4 サンプルの振幅

と縦軸のラベルは，それぞれ **xlabel 関数**と **ylabel 関数**により表示している。

余談であるが，MATLAB でプロットした図のフォントのデフォルトの大きさは，2 カラムの論文に掲載することを想定すると小さい。フォントサイズは毎回修正することもできるが，MATLAB 起動時に自動的に実行されるスクリプトである **startup.m** を活用すれば，デフォルトのフォントサイズを変更できる。具体的には，MATLAB 起動時に

```
>> edit startup.m
```

とコマンドを打ち，エディタで

```
set(0,'DefaultAxesFontSize',14);
set(0,'DefaultTextFontSize',14);
```

と打ち込んで保存する。コマンドウィンドウに入力しないため，>> は付していない。その後，MATLAB を再起動すれば，図のフォントサイズがつねに 14 で表示される。この処理は，1 回行えば以後恒久的に利用できる。MATLAB 起動時のフォルダを変更した場合は，同様の処理を再度行えばよい。

波形を表示する際には，横軸は時間（秒やミリ秒），縦軸は振幅にすることが推奨される。振幅は，A-D 変換時のマイクボリュームによって同じ大きさの音でも離散信号としては異なる値になるため，単位を記載しない。具体的には

```
>> xlabel('Time (s)');
>> ylabel('Amplitude');
```

とする。英語で記載する際，単語と括弧や単位との間に半角スペースを空ける必要があることに注意しよう。本書では図のキャプションを日本語にしているが，論文に記載することを意識してここでは英語で記載している。

最後に，本書における数式の表現方法について記載する。A-D 変換前の信号は $x(t)$ とし，A-D 変換後の信号は $x[n]$ と表すことにする。この二つの信号は

$$x[n] = x(nT) \tag{1.1}$$

という関係で結びつく。角括弧は，括弧の中身が整数の場合に使うことで離散

信号であることを明確に示す。また，離散信号の表現では，一部，数列を表現する際に用いる表記として x_n を用いることがある。この使い分けはほかの教科書等との整合性を意識しており，本書において数学的な意味での差はない。

1.2　ベクトルと内積

ある MATLAB の変数が N サンプルの振幅の時系列を持つ場合，N 次元の**ベクトル**（vector）であると表現する。MATLAB では行列演算を簡単にできるよう，配列では**横ベクトル**（row vector）と**縦ベクトル**（column vector）を区別する。例えば，以下のプログラムにおいて，x は横ベクトル，y は縦ベクトルとして格納される。

```
>> x=[1,3,-5,2];
>> y=[1;3;-5;2];
```

二つのベクトルの次元数が同じであれば，つぎの式により**内積**（inner product）が計算できる。プログラムと数式ではフォントが変わっているが，同じアルファベットであるものはプログラムと数式が対応する。

$$\vec{x}\cdot\vec{y}=\sum_{n=0}^{N-1}x[n]y[n] \tag{1.2}$$

ここで，$\vec{x}=(x[0],x[1],x[2],\cdots,x[N-1])$ とし，$\vec{y}$ も同様に N 次元のベクトルとする。この数式では行列の縦横は区別せず，$x[n]$ はベクトルの n 番目の要素を参照している。$\vec{x}$ の記載は教科書により異なることもあるが，本書では上記の書き方を用いる。また，内積は以下の式で定義されることもある。

$$\vec{x}\cdot\vec{y}=|\vec{x}|\,|\vec{y}|\cos\theta \tag{1.3}$$

表記の差はあれど意味するところは同じであるため，見かけの記述に惑わされずに理解してほしい。

内積を MATLAB で計算する場合，いくつかの選択肢がある。一番直接的な方法は，for 文を使い総和記号をそのまま実装するやり方であろう。ここでは，

索　　引

―― 著 者 略 歴 ――

2002年　釧路工業高等専門学校情報工学科卒業
2004年　和歌山大学システム工学部デザイン情報学科卒業
2006年　和歌山大学大学院システム工学研究科博士前期課程修了
2008年　和歌山大学大学院システム工学研究科博士後期課程修了（短期修了），博士（工学）
2008年　関西学院大学博士研究員
2009年　立命館大学助教
2013年　山梨大学特任助教
2017年　山梨大学准教授
2019年　明治大学准教授
　　　　現在に至る

ひたすら楽して音響信号解析
—MATLABで学ぶ基礎理論と実装—
Beginning Acoustic Signal Analysis
— Fundamental Theory and Its Implementation with MATLAB —

2021 年 2 月 12 日　初版第 1 刷発行　★

検印省略

著　者　森　勢　将　雅
発 行 者　株式会社　コ ロ ナ 社
　　　　　代 表 者　牛 来 真 也
印 刷 所　三 美 印 刷 株 式 会 社
製 本 所　有限会社　愛 千 製 本 所

112–0011　東京都文京区千石 4–46–10
発 行 所　株式会社　コ ロ ナ 社
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　https://www.coronasha.co.jp

ISBN 978–4–339–00939–2　C3055　Printed in Japan　（松岡）