

日本音響学会 編

音響入門シリーズ B-4

デジタル音響信号処理入門

－Python による自主演習－

博士(工学) 小澤 賢司 著

コロナ社

音響入門シリーズ編集委員会

編集委員長

大川 茂樹（千葉工業大学）

編集委員（五十音順）

小澤 賢司（山梨大学） 鈴木 陽一（東北文化学園大学）
羽田 陽一（電気通信大学） 誉田 雅彰（早稲田大学名誉教授）
柳田 益造（同志社大学名誉教授）

（2022年8月現在）

刊行のこ と ば

われわれは、さまざまな「音」に囲まれて生活している。音楽のように生活を豊かにしてくれる音もあれば、騒音のように生活を脅かす音もある。音を科学する「音響学」も、多彩な音を対象としており、学際的な分野として発展してきた。人間の話す声、機械が出す音、スピーカから出される音、超音波のように聞こえない音も音響学が対象とする音である。これらの音を録音する、伝達する、記録する装置や方式も、音響学と深くかかわっている。そのために、「音響学」は多くの人に興味をもたれながらも、「しきいの高い」分野であるとの印象をもたれてきたのではないだろうか。確かに、初心者にとって、音響学を系統的に学習しようとすることは難しいであろう。

そこで、日本音響学会では、音響学の向上および普及に寄与するために、高校卒業生・大学1年生に理解できると同時に、社会人にとっても有用な「音響入門シリーズ」を出版することになった。本シリーズでは、初心者にも読めるように想定されているが、音響以外の専門家でも、新たに音響を自分の専門分野に取り入れたいと考えている研究者や技術者も読者対象としている。

音響学は学際的な分野として発展を続けているが、音の物理的な側面の正しい理解が不可欠である。そして、その音が人間にどのような影響を与えるかも把握しておく必要がある。また、実際に音の研究を行うためには、音をどのように計測して、制御するのを知っておく必要もある。そのための背景としての各種の理論、ツールにも精通しておく必要がある。とりわけ、コンピュータは、音響学の研究に不可欠な存在であり、大きな潜在性を秘めているツールである。

このように音響学を学習するためには、「音」に対する多角的な理解が必要である。本シリーズでは、初心者にも「音」をいろいろな角度から正しく理解

ii 刊 行 の こ と ば

していただくために、いろいろな切り口からの「音」に対するアプローチを試みた。本シリーズでは、音響学にかかわる分野・事象解説的なものとして、「音響学入門」, 「音の物理」, 「音と人間」, 「音と生活」, 「音声・音楽とコンピュータ」, 「楽器の音」の6巻, 音響学的方法にかかわるものとして「デジタルフーリエ解析 (I) 基礎編, (II) 上級編」, 「電気の回路と音の回路」, 「デジタル音響信号処理入門 - Python による自主演習 -」の4巻 (計10巻) を継続して刊行する予定である。各巻とも、音響学の第一線で活躍する研究者の協力を得て、基礎的かつ実践的な内容を盛り込んだ。

本シリーズでは、CDやDVD、またはWebサイトに各種の音響現象を視覚・聴覚で体験できるコンテンツを用意している。また、読者が自己学習できるように、興味を持続させ学習の達成度が把握できるように、コラム (歴史や人物の紹介)、例題、課題、問題を適宜掲載するようにした。とりわけ、コンピュータ技術を駆使した視聴覚に訴える各種のデモンストレーション、自習教材は他書に類をみないものとなっている。執筆者の長年の教育研究経験に基づいて制作されたものも数多く含まれている。ぜひとも、本シリーズを有効に活用し、「音響学」に対して系統的に学習、理解していただきたいと願っている。

音響入門シリーズに飽きたらず、さらに音響学の最先端の動向に興味をもたれたら、日本音響学会に入会することをお勧めする。毎月発行する日本音響学会誌は、貴重な情報源となるであろう。学会が開催する春秋の研究発表会、分野別の研究会に参加されることもお勧めである。まずは、日本音響学会のホームページ (<https://acoustics.jp/>) をご覧になっていただきたい。

2022年8月

一般社団法人 日本音響学会 音響入門シリーズ編集委員会
編集委員長

まえがき

本書は、これから音響信号処理のプログラミングに取り組もうと考えている方を対象に、教師代わりに学習をナビゲートすることを意図した演習書です。想定読者は、音響学に関する研究室に配属された理工系学科の卒業研究生や修士1年生、そして音響関連のソフトウェア製品を開発する部署に新たに配属された社会人の方です。

多くの理工系学科では、基礎科目においてフーリエ級数展開、畳込み演算などを学ぶでしょう。しかし、プログラミングにまで踏み込むカリキュラムは多くないように思います。著者の所属学科でも同様に、卒業研究生の中には「音に興味があってこの研究室を選んだけれど、これまで信号処理はまったく学んだことがありません」という方がいます。本書は、まさにそのような方々に、「音の信号処理は案外簡単ですね」ということを実感していただくことを目的としています。そして読後には、自信をもってご自身の業務に臨んでいただければ幸いです。


さて、本来ならば講義科目で信号処理の原理を学び、その後でプログラミング演習を行うのが望ましいと思われれます。一方、本書は「離散フーリエ変換(DFT)の原理は知らなくても、与えられた音データに対するDFTの結果であるスペクトルを正しく理解して、必要な処理を実施できればよい」というスタンスです。すなわち、ブラックボックス化できる部分はそのままにして、音響信号処理全体を一通り完遂する力を習得することを目指しています。

本書は、対話的なプログラミング環境を用いることで、あたかも生徒が教師と対面で会話している雰囲気での学習を進めることを特徴とします。そのため、全体が「話し言葉に近い敬体」で書かれています。

また、本書では、その雰囲気を実現するために、プログラミング言語 Python[†] の対話的プログラミング環境である Google Colaboratory (略称: Colab) を用います。Python は、汎用性の高いプログラミング言語であり、コードがシンプルで扱いやすいように設計されているので、最近は人気の高い言語です。

そして、Colab では、テキストとプログラムコードを交互に配置することが可能なノートブックというインタフェースを利用するので、教師からの指導を受けて生徒がプログラムを実行するという疑似的な対面学習が可能です。特に重要な事項は枠で囲うことで明示しましたので、復習に利用してください。

本書の Python コードは、サポートページからダウンロードできますので、書籍中で「演習」とした課題については、ご自身でパラメータの値を変えて実行することをおすすめします。なお、サポートページの Colab ノートブックには、本書に掲載できなかった確認課題とその解答例や、やや高度な内容のコンテンツも示していますので、ぜひ一度サポートページをご覧ください。

プログラムを実行し、波形を目で見て・音として聞いてみると、「波形がこんなに違うのに、音色はほぼ同じ」ことや「波形の相違はわずかなのに、音色はかなり違う」ことを経験されるでしょう。これが聴覚の不思議なところであり、音響学については「百見は一聞にしかず」であることを実感されるでしょう。本書では、実際に音を聞くことができる箇所にをつけてあり、音を聞くためのサポートページを各章の冒頭で2次元コードにより案内しています。

なお、本書では、ディスプレイで出力を確認することを前提としていたので、図はカラーであることを想定していました。出版にあたりモノクロ図面としてありますことをご容赦ください。サポートページには、出力である音に加えて、図のカラー版も掲載しています。

読者の皆さまが、本書を通じてデジタル音響信号処理の歓びを実感していただければ幸いです。そして、本書ではブラックボックス化している原理的な部分まで、ご自身で興味をもって調べていただければ幸いです。そのための参

[†] 本書に記載の会社名、製品名は一般に各社の商標（登録商標）です。本文中では TM, ®マークは省略しています。

考書を、本書の引用・参考文献リストのページで紹介しています。

末筆ながら、本書の出版に関してご助力いただいた皆さまに感謝いたします。本シリーズ編集委員の鈴木陽一先生には、著者の研究室の卒業研究生用に開発した教材ノートブックを、本書として出版するようお声がけいただきました。同委員長の大川茂樹先生と同委員の羽田陽一先生には、原稿を通して読んでいただき、貴重なご意見をいただきました。コロナ社には、著者の遅筆をご寛容いただき、出版までご助力いただきました。また、塩澤光一朗様をはじめとする山梨大学の卒業研究生諸氏には、この教材を用いた学習の過程でさまざまなコメントをいただきました。重ねて感謝いたします。

2022年8月 コロナ禍の終息を祈りつつ

小澤賢司

■本書のサポートページ■

本書のサポートページは下記2次元コード、もしくはURLから見るができます。

https://kenjiozawa.github.io/DSP_practice_Python.html

(短縮URL: <https://bit.ly/3szss0H>) (2022年8月現在)



目 次

1. 演習環境の立上げ

1.1 Python と Colab	1
1.1.1 Python とは	1
1.1.2 Colab とは	1
1.2 演習環境の準備	2
1.2.1 Python 入門と Colab の利用	2
1.2.2 本書のサポートページ	2
1.3 Colab で WAV ファイルを聞いてみる	3

2. 音に触れる

2.1 音を数式で表現する	4
2.1.1 時間の関数としての音の数式表現	5
2.1.2 振幅と位相を含んだ数式表現	7
2.2 正弦波を生成して聞いてみる	9
2.3 A-D 変換について確認する	11
2.3.1 標 本 化	12
2.3.2 不適切な標本化	13
2.3.3 量 子 化	15
2.3.4 よろしくない量子化	17
2.3.5 量子化誤差を聞いてみる	19

3. アナログ音の周波数分析

3.1 正弦波の重ね合わせによる周期波形の合成	24
3.1.1 ウォーミングアップ：2成分複合音の合成	24
3.1.2 3成分複合音の合成	26
3.1.3 成分音の振幅と波形の関係	29
3.1.4 成分音の位相と波形の関係	30
3.1.5 三角関数の合成	32
3.2 フーリエ級数展開	33
3.2.1 丸い波による角のある波の合成	33
3.2.2 周期波形の分解	38
3.2.3 分解された成分の位相と振幅を調べる	41
3.3 スペクトル	42
3.3.1 フーリエ級数展開で得たスペクトルを描画する	42
3.3.2 実フーリエ級数展開から複素フーリエ級数展開へ	45
3.3.3 複素フーリエ級数展開で得たスペクトルを描画する	47
3.3.4 基本的な周期波形のスペクトル	50

4. デジタル音の周波数分析

4.1 デジタル信号のフーリエ変換	54
4.1.1 DFT と FFT	54
4.1.2 FFT スペクトルを読み解く（王道の解釈）	57
4.1.3 FFT スペクトルを読み解く（曲がった解釈）	60
4.1.4 スペクトルの振幅値に関する解釈	63
4.1.5 デジタル信号のスペクトル	64
4.2 窓関数	67
4.2.1 方形波窓による切出しが引き起こす問題	67
4.2.2 窓関数を用いた波形の切出し	71
4.2.3 STFT とスペクトログラム	75
4.3 窓関数による波形の変化とスペクトルの変化	80
4.3.1 切り出した波形の合成による元の波形の復元	80
4.3.2 窓関数の掛け算によるスペクトルの変化	84

4.4	忘れてはいけない位相スペクトル	87
-----	-----------------	----

5. 音のフィルタリング

5.1	FIR フィルタによる雑音の除去	92
5.1.1	インパルス応答の畳込み	92
5.1.2	FIR フィルタを自作する	96
5.1.3	FIR フィルタの自動設計 (窓関数法)	105
5.2	IIR フィルタによる雑音の除去	107
5.2.1	IIR フィルタの自動設計	107
5.2.2	IIR フィルタを利用した雑音除去	110
5.3	頭部伝達関数を用いた音像定位の制御	111
5.3.1	頭部伝達関数とは	111
5.3.2	頭部伝達関数を用いて音像を制御してみよう	112
5.4	FFT を利用した長い音のフィルタリング	117

6. さまざまな音響信号処理

6.1	ボイスチェンジャ	118
6.1.1	まずボイスチェンジしてみる	118
6.1.2	スペクトルの変化を観察する	121
6.1.3	スペクトルの変化を検討する	124
6.2	エフェクタ	127
6.2.1	トレモロ	127
6.2.2	ビブラート	130
6.3	マイクロホンアレイによるビームフォーミング	135
6.3.1	遅延和法 (DS 法) の基礎	135
6.3.2	遅延和法の実行	138

引用・参考文献	144
---------	-----

索引	145
----	-----

1

演習環境の立上げ

本書は、実際のプログラミングを通じて、音のデジタル信号処理を体得することを目的としています。本章では、その準備として、ブラウザから **Python** を実行できるサービスである **Google Colaboratory** (略称：**Colab**) を利用し、音を聞くことまでを実現しましょう。

1.1 Python と Colab

1.1.1 Python とは

汎用性の高いプログラミング言語である Python^{1)†} は、コードが読みやすく、さまざまな高性能ライブラリを無料で利用できることから、近年、注目を集めています。本書のプログラムは、Python 3.7 を用いて動作を確認しています。

1.1.2 Colab とは

Colab²⁾ は、Web ブラウザから Python プログラムを記述、実行できる無料サービスです。自身でプログラミング環境の設定をする必要がないので、すぐに Python プログラミングを体験できます。

Colab では、対話的コンピューティングを行うためにノートブック（ファイルの拡張子は .ipynb）と呼ばれるインタフェースを利用します。ノートブックでは、プログラムコード、数式を含む文章、グラフ、そして音の出力といったマルチメディアを混在させることができます。本書では、この機能を利用す

† 肩付き数字は巻末の引用・参考文献を示します。

ることで、「教師と対面で会話する感覚」で学習を進めることにします。

1.2 演習環境の準備

1.2.1 Python 入門と Colab の利用

本書では、何らかのプログラミング言語を習得した経験のある方であれば、Python を使ったことのない方でも読み下せる程度の平易なプログラミングを心がけたつもりです。もし Python に興味があれば、例えば Python 情報サイト¹⁾で入門記事を読むことをオススメします。なお、2022 年 5 月現在、このサイトにおける「ゼロからの Python 入門講座」では Colab を利用しています。そして、Colab を利用するために必要な、Google アカウント（無料）の作成方法から解説されています。

すでに Google アカウントを所有している方は、まず Colab を利用してみましょう。日本語の「Colaboratory へようこそ²⁾」にアクセスするだけです。ノートブックインタフェースにより、「解説を読む → コードを実行する」の繰り返しで、対話的に Python プログラミングを体験できます。

1.2.2 本書のサポートページ

本書で利用するファイル一式については、右の 2 次元コードで示す本書のサポートページ[†]に示してあります。本書における各章の記述のうち、コード部分だけを残した Colab ノートブックを提供していますので、ぜひ実際に手を動かしながら演習に取り組んでいただければ幸いです。また、サポートページの Colab ノートブックには、本書に掲載されていないコンテンツも含まれています。ぜひ、サポートページの目次情報をご覧ください幸いです。



さらには、サポートページの Colab ノートブックには、本書では割愛した「各節の内容に関する確認課題」が含まれています。また、その略解を Colab

[†] <https://bit.ly/3szss0H>

ノートブックとして提供しています。

1.3 Colab で WAV ファイルを聞いてみる

WAV ファイルは、Windows OS でよく利用されるオーディオデータのファイル形式です。本書のサポートページからダウンロードできる Colab ノートブックでは、必要なライブラリの `import` を各章の冒頭で済ませますので、以下の 1 行のプログラムで `sample/sample.wav` を聞くことができます。

```
In[ ]:  
Audio('sample/sample.wav')
```

```
Out[ ]:
```



三角形の「再生ボタン (▶)」を押せば音が再生されますし、音量も調整できます。

実行後の音は右の 2 次元コードのページで聞くことができます (サポートページからもアクセス可能です)。

以上で準備は完了です。次章以降では、Python による音の信号処理に徐々に慣れていきましょう。



2

音に触れる

本章では、最も単純かつ重要な波形である「正弦波」を計算機で生成することから始めます。そして、アナログ-デジタル変換（A-D変換）について学びます。

なお、本章において出力される音は、右の2次元コードのページでまとめて聞くことができます。



2.1 音を数式で表現する

今後の作業に備えて、波形を描く関数 `plot_wave` をここで定義しておきます（関数 `plot_wave` は、3章以降では徐々に機能を増やしていきます。ここでは、最も基本的な部分だけを示します）。

```
In[ ]:
def plot_wave(time, amplitude, xtitle = 'Time (s)', ¥
              ytitle = 'Amplitude (arb.)', hold = False, ¥
              color = 'blue', marker = ',', legend = '', ¥
              linestyle = '-'):
    ''' 時間軸データと波形データを受け取り、波形を描く関数を定義する
引数 time:      時刻の離散データ
    amplitude: 上記の時刻データに対応する瞬時振幅値
    xtitle:     横軸のラベル（暗黙値は 'Time (s)')
    ytitle:     縦軸のラベル（暗黙値は 'Amplitude (arb.)')
    hold:       False（暗黙値）ならば描画し、True ならばデータを保持する
```

```

color:      グラフの色（暗黙値はブルー）
marker:     マーカの種類（暗黙値は pixel）
legend:     凡例の文字列（暗黙値は「なし」）
linestyle:  線の種類（暗黙値は実線）
...

if (marker != ','): # マークが pixel でない場合には,
    linestyle = '' # データのプロットのみを行い, 線では結ばない

plt.plot(time, amplitude, color = color, marker = marker, ￥
         linestyle = linestyle, label = legend)
# この1行で描画される

if (legend != ''):
    # 凡例のデータが渡された場合には, 凡例を書く関数を呼ぶ
    plt.legend()

if (hold == False):
    # hold == False のときは, plt.show() を呼んで描画する
    plt.xlabel(xtitle)
    plt.ylabel(ytitle)
    plt.show()

```

2.1.1 時間の関数としての音の数式表現

高校の物理では、「音が伝搬する様子」を観察したでしょう（横軸は位置 x のグラフでした）。一方、本書では、「時間の関数としての音」の表現を考えていきます（横軸は時刻 t のグラフです）。音が伝搬しているときに、ある位置においては「時間とともに圧力が変化」しています。もし、その位置に耳があれば、「時間とともに鼓膜が押される／引かれる」が繰り返されるので、音として聞こえるのです。その位置にマイクロホンを置いたと考えると、100 Hz の音を 0.1 s 間だけ録音したのが **アウトプット 2.1** (図 2.1) の波形です (インプット 2.1 のプログラムにより描きました)。

索 引

<p style="text-align: center;">〔い〕</p> <p>位 相 7</p> <p>位相スペクトル 42, 46</p> <p>インパルス 87</p> <p>インパルス応答 93</p> <p style="text-align: center;">〔え〕</p> <p>エフェクタ 127</p> <p style="text-align: center;">〔お〕</p> <p>オイラーの定理 45</p> <p>オーバーラップアド 80</p> <p>音 像 111</p> <p>音像定位 111</p> <p style="text-align: center;">〔か〕</p> <p>角周波数 7</p> <p style="text-align: center;">〔き〕</p> <p>奇関数 49</p> <p>基本周波数 26</p> <p>基本周波数成分 26</p> <p>基本波 26</p> <p>逆離散フーリエ変換 60</p> <p style="text-align: center;">〔く〕</p> <p>空間折返しひずみ 136</p> <p>偶関数 49</p> <p>矩形波 36</p> <p style="text-align: center;">〔こ〕</p> <p>高速フーリエ変換 54</p> <p>高調波複合音 26</p>	<p style="text-align: center;">〔さ〕</p> <p>三角関数の直交性 41</p> <p style="text-align: center;">〔し〕</p> <p>周 期 7</p> <p>周波数 7</p> <p>周波数ビン 64</p> <p>周波数分解能 64</p> <p>純 音 6</p> <p>振 幅 7</p> <p>振幅スペクトル 42, 46</p> <p style="text-align: center;">〔す〕</p> <p>スペクトログラム 77</p> <p style="text-align: center;">〔せ〕</p> <p>線形位相 89</p> <p>線状畳込み 117</p> <p style="text-align: center;">〔た〕</p> <p>第 n 高調波 26</p> <p>第 n 倍音 26</p> <p>畳込み 93</p> <p>タップ長 97, 105</p> <p>単位パルス 87</p> <p>短時間フーリエ変換 74</p> <p style="text-align: center;">〔ち〕</p> <p>遅延和法 135</p> <p>重畳加算 80</p> <p>直線位相 89</p> <p>直線畳込み 117</p>	<p style="text-align: center;">〔と〕</p> <p>頭外定位 116</p> <p>頭内定位 115</p> <p>頭部インパルス応答 112</p> <p>頭部伝達関数 112</p> <p>トレモロ 127</p> <p style="text-align: center;">〔は〕</p> <p>ハニング窓 71</p> <p style="text-align: center;">〔ひ〕</p> <p>ビブラート 130</p> <p>標本化 11</p> <p>標本化周期 9</p> <p>標本化周波数 9</p> <p>標本化定理 12</p> <p style="text-align: center;">〔ふ〕</p> <p>フォルマント 56</p> <p>複素正弦波 45</p> <p>複素フーリエ級数展開 45</p> <p>負の周波数 46</p> <p>フーリエ級数展開 33</p> <p style="text-align: center;">〔ほ〕</p> <p>ボイスチェンジャ 118</p> <p>方形波 36</p> <p style="text-align: center;">〔ま〕</p> <p>窓関数 71</p> <p>窓関数法 100</p> <p style="text-align: center;">〔り〕</p> <p>離散フーリエ変換 54</p>
---	--	---

量子化 11 | 量子化誤差 18 | 量子化精度 16



	[A]		[F]	HRTF	112
A-D 変換	11	FFT	54	[I]	
	[C]	FIR	92, 97	IDFT	60
Colab	1		[G]	IIR	92, 108
	[D]	Gibbs 現象	37	[P]	
DFT	54	Google Colaboratory	1	Python	1
			[H]	[S]	
		HRIR	112	STFT	74

— 著者略歴 —

小澤 賢司 (おざわ けんじ)

1986年 東北大学工学部通信工学科卒業
1988年 東北大学大学院工学研究科博士前期課程修了 (電気及通信工学専攻)
1988年 東北大学助手
1994年 博士 (工学) (東北大学)
1998年 東北大学助教授
1998年 山梨大学助教授
2007年 山梨大学教授
現在に至る

デジタル音響信号処理入門

— Python による自主演習 —

Introduction to Digital Acoustic Signal Processing

— Independent Exercises in Python —

© 一般社団法人 日本音響学会 2022

2022年10月7日 初版第1刷発行

検印省略

編者 一般社団法人 日本音響学会
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 新日本印刷株式会社
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 978-4-339-01310-8 C3355 Printed in Japan

(森)



本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。
購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。
落丁・乱丁はお取替えいたします。