

# スパースモデリング

— 基礎から動的システムへの応用 —

博士(情報学) 永原 正章 著

コロナ社

# ま え が き

本書はスパースモデリング (sparse modeling) の基礎とその動的システムへの応用について、わかりやすく解説した専門書である。大学の初年度に習う線形代数の知識と数値計算ソフトウェア MATLAB の使い方を知っていれば、スパースモデリングの基本的な考え方から、最新の研究成果である動的スパースモデリングの計算まで、短期間で学ぶことができるだろう。特に動的スパースモデリングに関する書籍は、世界的に見ても本書が初である。

理工系で成功するための最も重要なスキルは、情報を受信するアンテナの感度をつねに高くして、国内外でどのような理論や技術が話題になりつつあるかを捉えることである。現在、データ科学や人工知能が全世界的なブームであることは、そのようなアンテナがなくても容易にわかるだろう。しかし、もっと注意深くこのブームの背景を見てみると、スパース性に関する技術が大きく関与していることに気付く。スパースモデリングやスパースコーディング (sparse coding), スパース最適化 (sparse optimization), 圧縮センシング (compressed sensing) などがスパース性に関する技術である。アンテナ感度の高い研究者やエンジニアたちは、ブームになる前にこのような技術に着目し、いち早く自分の研究開発分野に取り込んで、先取権を獲得する。世の中の科学技術の多くはこのようにして発展してきた。スパースモデリングの産業応用はまだ始まったばかりである。ぜひ本書でスパースモデリングを勉強して、あなたの分野に新しい研究・開発の風を吹かせてほしい。

最近の人工知能の発展 (第3次ブーム) により、人工知能はたいへん身近な存在となった。特に、音声および画像の認識や分類、ビッグデータ解析、IoT (Internet of Things) の枠組みにおけるエッジコンピューティング、フォグコンピューティングなど、まさにわれわれの周りを取り巻く環境に人工知能の技

術が自然に導入され始めている。そのような人工知能の多くは、「ものの理解」や「見える化」を目指している。いわば、超高機能なセンサを作ろうとしているといえる。一方、人工知能の分析結果を基に環境に働きかけ、スマートにものを動かすためには、超高機能なアクチュエータも欠かせない。そのための基礎理論は自動制御理論と呼ばれる。スマートなセンサ（人工知能）とスマートなアクチュエータ（自動制御）が組み合わさって初めて、スマートに動くものを作ることができる。

このような背景から、スパースモデリングと自動制御理論を組み合わせた研究が最近注目されており、これを動的スパースモデリングと呼ぶ。動的スパースモデリングでは、燃料や電力の消費、CO<sub>2</sub>の排出、騒音や振動の発生など環境に悪影響を及ぼす要因を制約条件とした最適化問題として制御問題を定式化する。これらを陽に考慮した制御系を設計することが可能となり、まさに、省エネルギーを達成するための環境にやさしい自動制御理論となっている。本書で学んだ動的スパースモデリングを活用して、さまざまな環境問題を解決するヒントをぜひ得ていただきたい。

本書のおもな目標は、スパースモデリングの技術を使うことにあるので、重要ではあるが幾分難しいスパースモデリングの理論的な側面、例えばランダム行列理論を用いたスパース最適解の特徴付けや各種アルゴリズムの収束性の議論などは大きく省略した。これらの理論的な側面に興味がある方は、各章の終わり（5章を除く）の「さらに勉強するために」と題された節で挙げられている文献にチャレンジしてほしい。

本書の構成は下記のとおりである。

- スパース性とは何か（1章）… 最初の章では、スパース性の数学的な定義を述べる。本書を通じて大切な章である。
- 曲線フィッティングで学ぶスパースモデリング（2章）… データからの曲線フィッティングを題材に最小二乗法や正則化、 $\ell^1$ 最適化などを学ぶ。また、これらの最適化問題を解くためのMATLABの使い方もここで解説する。

- 凸最適化アルゴリズム (3章) ... スパースモデリングで重要な技術である凸最適化のアルゴリズムを解説する。
- 貪欲アルゴリズム (4章) ... スパースモデリングで凸最適化と並んで重要な貪欲アルゴリズムについて解説する。
- スパースモデリングの歴史 (5章) ... コーヒーブレイクとしてスパースモデリングの歴史について振り返る。省略可能。
- 動的システムと最適制御 (6章) ... 動的スパースモデリングに必要な動的システムと最適制御の基礎について勉強する。
- 動的スパースモデリング (7章) ... 動的システムに対するスパースモデリング (動的スパースモデリング) の基礎を勉強する。
- 動的スパースモデリングのための数値最適化 (8章) ... 動的スパースモデリングの最適化問題を解くための数値最適化手法について示す。

本書の一つの特徴は、スパースモデリングを実際に試してみるための MATLAB コードを掲載していることである。本書を読み進めるとともに、MATLAB でいろいろな数値例題を試してみることで、スパースモデリングの深い理解が可能になる。

本書に掲載している MATLAB コードや追加情報については以下のサポートページを参照してほしい。

<https://nagahara-masaaki.github.io/spm.html>

本書の内容は、科研費新学術領域「スパースモデリングの深化と高次元データ駆動科学の創成」や計測自動制御学会 (SICE)「モデルベースト制御における機械学習とダイナミクスの融合」調査研究会、ひびきの AI 社会実装研究会などでのセミナーや交流会、また多くの研究者・エンジニアとのディスカッションで勉強させていただいた内容が多く含まれている。また、本書の内容の一部 (特に 6 章以降) は JSPS 科研費 15H02668 および 16H01546 の助成を受けた研究成果に基づいている。

九州工業大学名誉教授の石川真澄先生には、ニューラルネットワークについての先生のご研究に関する文献と貴重なコメントをいただいた。大阪大学工学研

究科の林直樹先生と株式会社 IHI の濱口謙一氏，また京都大学情報学研究科の藤本悠介氏には，本書の草稿について詳細なコメントをいただいた。京都大学病院の藤本晃司先生からは MRI に関して貴重なコメントをいただいた。コロナ社には本書の出版にあたりたいへんお世話になった。本書の執筆・出版を支えていただいた多くの方々に感謝の意を表したい。

2017 年 9 月

著 者

コロナ社

# 目 次

## 1. スパース性とは何か

1.1 冗長な辞書とスパース性	2
1.2 $\ell^0$ ノルムの定義と意味	5
1.3 総当り法による解法	9
1.4 さらに勉強するために	12

## 2. 曲線フィッティングで学ぶスパースモデリング

2.1 最小二乗法と正則化	15
2.1.1 劣決定系と最小ノルム解	15
2.1.2 回帰問題と最小二乗法	19
2.1.3 正則化法	24
2.2 スパースモデリングと $\ell^1$ ノルム最適化	29
2.3 CVX による数値最適化	33
2.4 さらに勉強するために	37

## 3. 凸最適化アルゴリズム

3.1 凸最適化問題への準備	40
3.2 近接作用素	45
3.2.1 近接作用素の定義	45

3.2.2	近接アルゴリズム	47
3.2.3	2次関数の近接作用素	48
3.2.4	指示関数の近接作用素	49
3.2.5	$\ell^1$ ノルムの近接作用素	50
3.3	近接分離法による $\ell^1$ 最適化の数値解法	54
3.4	近接勾配法による $\ell^1$ 正則化の数値解法	57
3.5	一般化 LASSO と ADMM	62
3.6	さらに勉強するために	68

## 4. 貪欲アルゴリズム

4.1	$\ell^0$ 最適化	72
4.2	直交マッチング追跡	75
4.2.1	マッチング追跡 (MP)	75
4.2.2	直交マッチング追跡 (OMP)	80
4.3	しきい値アルゴリズム	83
4.3.1	反復ハードしきい値アルゴリズム (IHT)	84
4.3.2	反復 $\ell_1$ スパースアルゴリズム	85
4.3.3	圧縮サンプリングマッチング追跡 (CoSaMP)	88
4.4	数値実験	89
4.5	さらに勉強するために	92

## 5. スパースモデリングの歴史

5.1	オッカムの剃刀	97
5.2	グループテストイング	99
5.3	$\ell^1$ ノルムによる最適化	102

5.3.1 信号復元問題	102
5.3.2 地球物理学	103
5.3.3 ニューラルネットワーク	104
5.3.4 統計的学習	104
5.3.5 信号処理	105
5.4 自動制御とスパースモデリング	106

## 6. 動的システムと最適制御

6.1 動的システム	110
6.1.1 状態方程式	111
6.1.2 可制御性と可制御集合	113
6.2 最適制御	117
6.3 ロケットの最短時間制御	121
6.4 さらに勉強するために	128

## 7. 動的スパースモデリング

7.1 連続時間信号のノルムとスパース性	130
7.1.1 $L^p$ ノルム	130
7.1.2 $L^0$ ノルムとスパース性	132
7.2 スパースな制御の工学的な意義	133
7.3 動的スパースモデリングの定式化	135
7.4 $L^0$ 最適制御と $L^1$ 最適制御の等価性	138
7.4.1 ポントリャーギンの最小原理	138
7.4.2 正 規 性	139
7.4.3 定理7.1の証明	140

7.5	スパースモデリングとの関係	141
7.6	ロケットのスパース最適制御	143
7.7	離散値制御	148
7.7.1	絶対値和 (SOAV) 最適制御	148
7.7.2	ポントリャーギンの最小原理	150
7.8	さらに勉強するために	155

## 8. 動的スパースモデリングのための数値最適化

8.1	時間軸の離散化	158
8.2	有限次元最適化問題への帰着	160
8.3	ADMM による高速アルゴリズム	163
8.4	さらに勉強するために	168
引用・参考文献		169
演習問題解答		176
索引		206

# 1

## スパース性とは何か

本章では、有限次元のベクトルのスパース性を数学的に定式化する。本書を通じて、本章で説明する概念が重要となるので、しっかり勉強していただきたい。

本書では、有限次元のベクトル（有限次元空間上に矢印として書けるベクトル）を太文字で  $\boldsymbol{x}$  などと書くことにする。ただし、1次元ベクトルはスカラと同一視し、太文字では書かない。有限次元のベクトルは列ベクトル（縦ベクトル）であり

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (1.1)$$

のように表現する。ただし、本文中で有限次元ベクトルを書く場合は

$$\boldsymbol{x} = (x_1, x_2, x_3) \quad (1.2)$$

のように丸括弧で横書きする場合もあるが、これは縦ベクトル (1.1) に読み替えていただきたい。ベクトル  $\boldsymbol{x}$  の転置を  $\boldsymbol{x}^\top$  で表す。また、 $n$ 次元実ベクトル空間（ $n$ 次元の列ベクトルの空間）を  $\mathbb{R}^n$  で表す。

### 1章の要点

- スパースモデリングでは冗長な辞書（ベクトルの集合）を考える。
- スパースモデリングでは、冗長な辞書から最も少ないベクトルを選んで信号を表現する（ $\ell^0$ 最適化）。
- $\ell^0$ 最適化問題の総当り法による解法は、問題のサイズが大きくなれば指数関数的に計算量が増大する。

## 1.1 冗長な辞書とスパース性

3次元空間  $\mathbb{R}^3$  を考えよう。 $\mathbb{R}^3$  の標準基底 (standard basis) は

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (1.3)$$

である。この標準基底  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  を用いて、任意の3次元ベクトル  $\mathbf{y} \in \mathbb{R}^3$  は

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = y_1 \mathbf{e}_1 + y_2 \mathbf{e}_2 + y_3 \mathbf{e}_3 \quad (1.4)$$

と書くことができる。一般には、 $\mathbb{R}^3$  から独立な3本のベクトル  $\phi_1, \phi_2, \phi_3$  を持ってくれば、それらは  $\mathbb{R}^3$  の基底となる。すなわち、任意の  $\mathbf{y} \in \mathbb{R}^3$  に対して、ある実数  $\beta_1, \beta_2, \beta_3$  が一意に存在して

$$\mathbf{y} = \beta_1 \phi_1 + \beta_2 \phi_2 + \beta_3 \phi_3 \quad (1.5)$$

と表現できる。もし、 $\phi_1, \phi_2, \phi_3$  が長さ1で互いに直交する正規直交基底 (orthonormal basis) ならば、すなわち

$$\phi_i^\top \phi_j = \begin{cases} 1, & i=j \\ 0, & i \neq j \end{cases}, \quad i, j = 1, 2, 3 \quad (1.6)$$

が成り立てば

$$\beta_i = \phi_i^\top \mathbf{y}, \quad i = 1, 2, 3 \quad (1.7)$$

として、係数  $\beta_1, \beta_2, \beta_3$  が求まる。

---

**演習問題 1.1** ベクトル  $\mathbf{y} \in \mathbb{R}^3$  と正規直交とは限らない基底  $\phi_1, \phi_2, \phi_3$  が与えられたとき、表現 (1.5) の係数  $\beta_1, \beta_2, \beta_3$  を求めよ。

---

$\mathbb{R}^3$  の基底として

$$\phi_1 = e_1 + e_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad \phi_2 = e_2 + e_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \phi_3 = e_3 + e_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \quad (1.8)$$

を考えよう。これと標準基底 (1.3) とを合わせて、ベクトルの組  $\{e_1, e_2, e_3, \phi_1, \phi_2, \phi_3\}$  を作る。図 1.1 にこれら 6 本のベクトルを示す。これら 6 本のベクトルを用いて、3次元ベクトル  $y \in \mathbb{R}^3$  を

$$y = \sum_{i=1}^3 \alpha_i e_i + \sum_{i=1}^3 \beta_i \phi_i \quad (1.9)$$

と表現したとしよう。これは明らかに冗長な表現であり、式 (1.9) を満たす係数  $\alpha_i, \beta_i$  ( $i = 1, 2, 3$ ) の組合せは無数に存在する。例えば、 $y = (y_1, y_2, y_3)$  のとき、自明な一つの解として

$$(\alpha_1, \alpha_2, \alpha_3) = (y_1, y_2, y_3), \quad (\beta_1, \beta_2, \beta_3) = (0, 0, 0) \quad (1.10)$$

がある。

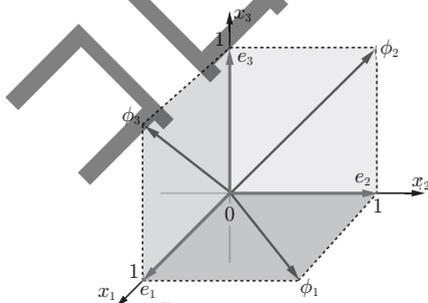


図 1.1  $\mathbb{R}^3$  の 6 本のベクトル  $e_1, e_2, e_3, \phi_1, \phi_2, \phi_3$

いま、非ゼロの係数を記憶するためのメモリがきわめて高価で、なるべく多くの係数を 0 にしたいという状況を考えよう。この状況の下で、ベクトル  $y \in \mathbb{R}^3$  が二つのベクトル  $e_1$  と  $\phi_2$  の張る平面上にあるとしよう。すると、 $\alpha_1$  と  $\beta_2$  以外の係数をすべて 0 にして

$$\mathbf{y} = \alpha_1 \mathbf{e}_1 + \beta_2 \phi_2 \quad (1.11)$$

という表現が可能となる。すなわち自明な係数 (1.10) よりも非ゼロの係数が少ない表現が得られたことになる。三つの非ゼロ要素が二つに減っただけでは、あまりご利益はないと思われるかもしれない。しかし、例えば 10 万次元のベクトルが 100 本のベクトルだけで表現できたとすると、大幅なデータ圧縮が可能となる。

以上の考察を一般化してみよう。 $m$  次元のベクトル空間  $\mathbb{R}^m$  を考える。そこに、次元よりも多いベクトル  $\{\phi_1, \phi_2, \dots, \phi_n\}$  を導入し (ただし,  $m < n$ )、ベクトル  $\mathbf{y} \in \mathbb{R}^m$  に対して

$$\mathbf{y} = \sum_{i=1}^n \alpha_i \phi_i \quad (1.12)$$

を満たす係数  $\alpha_1, \alpha_2, \dots, \alpha_n$  を求めたいとしよう。ただし、ベクトル  $\phi_i$  のうち  $m$  本は 1 次独立であるとする。このようなベクトルの組  $\{\phi_1, \phi_2, \dots, \phi_n\}$  を辞書 (dictionary) と呼ぶ。いまの場合、辞書のサイズ  $n$  はベクトル  $\mathbf{y}$  の次元  $m$  よりも大きい。このような辞書を冗長な辞書 (redundant dictionary) と呼ぶ。

行列  $\Phi$  とベクトル  $\mathbf{x}$  を

$$\Phi \triangleq \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_n \end{bmatrix} \in \mathbb{R}^{m \times n}, \quad \mathbf{x} \triangleq \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} \in \mathbb{R}^n \quad (1.13)$$

と定義すると、式 (1.12) の等式は

$$\mathbf{y} = \Phi \mathbf{x} \quad (1.14)$$

と簡単に書くことができる。いま、冗長な辞書による表現を考えているため、行列  $\Phi$  は横長の行列となっていることに注意する。スパースモデリングでは、このような横長の行列による連立方程式が考察の対象となる。

ベクトル  $\phi_i \in \mathbb{R}^m$  のうち  $m$  本は 1 次独立であると仮定したので、行列  $\Phi$  はフルランク (full rank) である。ここで、行列  $\Phi \in \mathbb{R}^{m \times n}$  がフルランクであるとは

$$\text{rank}(\Phi) = \min(m, n) \quad (1.15)$$

が成り立つことである。このとき、任意の  $\mathbf{y} \in \mathbb{R}^m$  に対して、線形方程式 (1.14) を満たす  $\mathbf{x}$  は少なくとも一つ存在する。その解の一つを  $\mathbf{x}_0$  としよう。つぎに行列  $\Phi$  の零化空間 (null space) またはカーネル (kernel) を

$$\ker(\Phi) \triangleq \{\mathbf{x} \in \mathbb{R}^n : \Phi \mathbf{x} = \mathbf{0}\} \quad (1.16)$$

で定義する。この  $\ker(\Phi)$  は線形空間 (すなわち、原点を通る  $n$  次元空間の超平面) である。線形代数の次元定理<sup>†</sup>により、 $\ker(\Phi)$  の次元は  $n - m$  となり、 $n > m$  であるので、 $\ker(\Phi)$  の元は無数にある。その元  $\mathbf{z} \in \ker(\Phi)$  を用いれば、線形方程式 (1.14) の一般解は

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{z}, \quad \mathbf{z} \in \ker(\Phi) \quad (1.17)$$

と表現できる。これより、方程式 (1.14) を満たす  $\mathbf{x}$  は無数に存在することがわかる。

**演習問題 1.2** 式 (1.17) のベクトル  $\mathbf{x}$  が方程式 (1.14) の解であることを示せ。

無数に存在する解の中から非ゼロの係数が最も少ないもの、言い換えれば最もスパースなものを選ぶのがスパースモデリングの基本問題である。次節以降でこの問題を定式化しよう。

## 1.2 $\ell^0$ ノルムの定義と意味

ここでは、有限次元ベクトルのノルムについて復習し、スパース性を定式化

<sup>†</sup> 次元定理 (dimension theorem): 行列  $\Phi \in \mathbb{R}^{m \times n}$  に対して、 $\ker(\Phi)$  の次元と  $\Phi$  の階数  $\text{rank}(\Phi)$  を足すと  $n$  になる。

# 索引

<b>【あ】</b>		<b>【き】</b>		<b>【さ】</b>	
悪条件	49	基底追跡	31, 105	最小エネルギー制御	143
圧縮サンプリングマッチング		軌道計画	113	最小二乗解	23, 81
追跡	88	軌道生成	108, 113	最小二乗法	22
圧縮センシング	105	逆行列補題	164	最小燃料制御	106, 137
<b>【い】</b>		逆問題	103	最小ノルム解	17
1次収束	80, 85, 88	共状態	120	最大値ノルム	7
一般化 LASSO	62	行フルランク	17	最短時間	118
<b>【う】</b>		局所最適解	44	最短時間制御	118
ヴァンデルモンド行列	20	曲線フィッティング	15	最適状態	120
ウェーブレット	12	極値制御	121	最適制御	119
運転曲線（鉄道の）	156	許容集合	43	三角不等式	6
<b>【え】</b>		切替え曲線	126	三重対角行列	65
枝刈り	89	近接アルゴリズム	47	<b>【し】</b>	
エピソード	42	近接勾配法	58	磁気共鳴画像法	70
エラスティックネット	105	近接作用素	45	次元縮約	13
<b>【お】</b>		近接分離アルゴリズム	55	次元定理	5
オッカムの剃刀	97	<b>【く】</b>		指示関数	49, 163
<b>【か】</b>		組合せ最適化	12, 102	辞書	4
カーネル	5	グリーンな制御	134	辞書学習	14
回帰分析	19	グループテスト	99	システム同定	103, 128
過学習	22, 104	グループ LASSO	105	実行可能解	43
拡張ラグランジュ関数	63	<b>【け】</b>		実行可能制御	117
可制御	114	経験モデリング	128	実行可能領域	43
可制御集合	116	ケチの原理	97	実効定義域	41
可制御性行列	114	<b>【こ】</b>		射影	46
可制御性グラミアン	194	交互方向乗数法	62, 163	射影作用素	50, 81
過適合	22	コーシー・シュワルツの不等式	177	縮小推定	104
		コスト関数	43	条件数	49
		ゴールドバーク機械	97	状態	111
				状態観測問題	108
				状態方程式	111
				冗長な辞書	4

初期状態 111  
 信号復元問題 102  
 深層学習 38, 104  
 深層ニューラルネットワーク 38

**【す】**

数値最適化 32  
 スパース (ベクトルが) 8, 30  
 スパース (連続時間信号が) 133  
 スパース最適制御 136  
 スペクトル半径 61

**【せ】**

正規 (最適制御問題が) 140, 154  
 正規直交基底 2  
 制御 111  
 制御する 112  
 制御対象 111  
 斉次性 6  
 正準方程式 120  
 正則化項 27  
 正則化最小二乗法 27  
 正則化パラメータ 27  
 正定値 48  
 制約集合 43  
 制約条件 43  
 絶対値和 150  
 絶対値和最適制御 150  
 0-1 最適化 102  
 零化空間 5  
 線形行列不等式 107  
 線形収束 80  
 全変動 66, 71, 105  
 全変動ノイズ除去 65

**【そ】**

総当り法 9, 74  
 相互コヒーレンス 73  
 双線形行列不等式 107

ソフトしきい値作用素 51, 165

**【た】**

台 (関数の) 132  
 台 (ベクトルの) 8, 193  
 帯域制限 102  
 大域的最適解 44  
 第一原理モデリング 128  
 ダグラス・ラシュフォード分離 55  
 多項式曲線フィッティング 19  
 多層パーセプトロン 104

**【ち】**

地球物理学 103  
 中線定理 182  
 直交マッチング追跡 80

**【て】**

ディープニューラルネットワーク 38  
 ディープラーニング 38  
 データ圧縮 13

**【と】**

等長制約性条件 92  
 動的システム 110  
 動的スパースモデリング 109, 136  
 動的モード分解 129  
 特異 (最適制御問題が) 140  
 特異区間 140  
 独立性 6  
 凸関数 41  
 凸最適化問題 31, 43  
 凸集合 40  
 ドロップアウト 38, 104  
 貪欲法 75

**【に】**

ニューラルネットワーク 104

**【ね】**

ネットワーク化制御系 108, 148

**【の】**

濃度 (有限集合の) 8  
 ノルム 6

**【は】**

ハードしきい値作用素 53, 84  
 ハミルトニアン 120  
 ハミルトンの正準方程式 121  
 バン・オフ・バン制御 106, 140  
 バン・バン制御 123  
 反復縮小しきい値アルゴリズム 61  
 反復ハードしきい値アルゴリズム 85, 87  
 反復  $s$ -スパースアルゴリズム 87

**【ひ】**

ビッグデータ 13  
 非凸関数 42  
 非凸集合 40  
 標準基底 2, 76

**【ふ】**

フィードバック制御 113, 127  
 フィードフォワード制御 112  
 フーリエ級数 142  
 不感帯関数 139  
 符号関数 123  
 不確かさ (制御対象の) 107  
 不動点 48, 59  
 不変集合 46  
 フレーム 12  
 プロパー 41

**【へ】**

閉関数 42



— 著者略歴 —

愛媛県生まれ。2003年、京都大学大学院情報学研究所博士課程修了。博士（情報学）。京都大学助手、助教、講師を経て、2016年より北九州市立大学環境技術研究所教授。また、同年よりインド工科大学ムンバイ校（IIT Bombay）の客員教授を兼任。専門は自動制御と人工知能。2012年、IEEE制御システム部門より国際賞である Transition to Practice Award を受賞。同賞の受賞は日本人初である。そのほか、計測自動制御学会や電子情報通信学会の論文賞など、受賞多数。IEEEの上級会員（Senior Member）。著書に「マルチエージェントシステムの制御」（コロナ社、共著）などがある。

スパースモデリング — 基礎から動的システムの応用 —

Sparse Modeling—Fundamentals and Its Applications to Dynamical Systems—

© Masaaki Nagahara 2017

2017年10月31日 初版第1刷発行

★

検印省略

著者 <sup>なが</sup>水 <sup>はら</sup>原 <sup>まさ</sup>正 <sup>あき</sup>章  
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 <http://www.coronasha.co.jp>

ISBN 978-4-339-03222-2 C3053 Printed in Japan

(横尾)



©COPY

< 出版者著作権管理機構 委託出版物 >

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-3513-6969, FAX 03-3513-6979, e-mail: info@jcopy.or.jp）の許諾を得てください。

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。