Geometric Semantic Genetic Programmingにおける ターゲットの意味を考慮した交叉の提案

○原章 串田淳一 種村涼 高濱徹行 （広島市立大学）

New Genetic Operator Considering Target Semantics in Geometric Semantic Genetic Programming

∗A. Hara, J. Kushida, R. Tanemura and T. Takahama (Hiroshima City University)

Abstract– In this paper, we focus on solving symbolic regression problems by Geometric Semantic Genetic Programming (GSGP). In GSGP, offspring is produced by a convex combination of two parental individuals. In order to improve the search performance of GSGP, we propose an improved Geometric Semantic Crossover utilizing the information of the target semantics. In conventional GSGP, ratios of convex combinations are determined at random. On the other hand, our proposed method can use optimal ratios for affine combinations of parental individuals. We confirmed that our method showed better performance than conventional GSGP in several symbolic regression problems.

Key Words: Genetic Programming, Evolutionary Computation, Symbolic Regression

1 はじめに

遺伝的プログラミング (Genetic Programming; GP)[1] はプログラム自動生成のための進化的計算手法である．通常，GP における個体は関数記号と終端記号を組み合わせた木構造プログラムで表現される．ランダムに生成した初期個体群に対して，選択・交叉・突然変異といった遺伝的操作を繰り返し適用し，集団を進化させる．

近年，プログラムの振る舞い（意味）に着目した新しい遺伝操作が研究されている．プログラムの意味は，例えば関数同定問題においては，与えられる入力の組に対する出力の組とみなすことができる．Moraglio らは，この意味に基づいた Geometric Semantic Genetic Programming (GSGP)[2, 3] を提案している．GSGP において子個体は，親個体の構造的な形質ではなく意味的な形質を継承するように生成される．

GSGP における交叉である Geometric Semantic Crossover では，子個体が 2 つの親個体の凸結合によって生成される．これは言い換えれば，意味空間において 2 つの親個体の内分点に相当する位置に子個体が生成されることを意味する．よってこの交叉では，子個体は適応度が低い方の親個体よりも適応度が改善するという特徴を持つ．しかし，2 つの親個体が張る意味空間の外側に目標とする解がある場合，両方の親個体の適応度を上回る子個体は生成できない．

ところで，関数同定問題は，あらかじめ与えられた入出力関係を満たす関数を導出する問題である．よって，意味空間上で最適解の位置があらかじめ与えられていると考えることができる．本研究では，この最適解の意味を子個体生成のターゲットとして陽に利用した，より効率の良い改良型の Geometric Semantic Crossover を提案する．いくつかの関数同定問題において提案する遺伝操作を導入した GSGP と従来の GSGP との性能比較を行い提案手法の有効性を検証する．

本論文の構成は下記のとおりである．2 節では，木構造プログラムの意味の定義とこれまでに提案された意味に基づく遺伝操作および GSGP について述べる．3 節では，本論文で提案するターゲットの意味を考慮した Geometric Semantic Crossover について説明する．4 節で実験結果と考察について述べ，5 節でまとめと今後の課題について述べる．

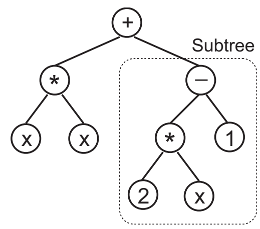


Fig. 1: An example of tree structural program for symbolic regression problems.

2 GP における意味に基づく交叉

2.1 木構造プログラムの意味

関数同定問題においては，(部分）木の意味は与えられる入力の組に対する対象の木の出力の組として定義される [4, 5]．木全体の意味はルートノードが出力する値の組に相当し，部分木の意味はその部分木の親ノードが受け取る値に相当する．例として，関数 $f(x)$ を同定する場合を取り上げ，Fig. 1 に示すような木構造プログラムを考える．ここでは，変数 x に対する入力 (fitness case) として，$x = 1, 2, 3, 4$ の 4 つのケースがあるものとする．各々の入力に対して，図中の点線で示した部分木 $(-\ (*\ 2\ x)\ 1)$ の出力は以下のようになる．

$$
\begin{aligned}
(-\ (*\ 2\ 1)\ 1) &= 1,\\
(-\ (*\ 2\ 2)\ 1) &= 3,\\
(-\ (*\ 2\ 3)\ 1) &= 5,\\
(-\ (*\ 2\ 4)\ 1) &= 7.
\end{aligned}
$$

よって，この部分木の意味は $(1, 3, 5, 7)$ となる．このように部分木の意味は，fitness case と同じ数の次元数を持つベクトルとみなすことができる．同様に，ルートノードが返す値を計算することにより，この木構造プログラム全体の意味は $(2, 7, 14, 23)$ と求まる．

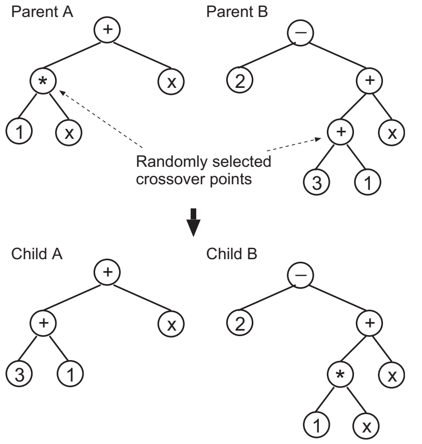


Fig. 2: Standard subtree exchange crossover in GP.

2.2 意味に基づく交叉

GP で一般的に利用される部分木交換交叉では，2 つの親個体から各々ランダムに交叉点が選択され，その交叉点をルートとする部分木を交換することにより子個体が生成される．Fig. 2 に部分木交換交叉の例を示す．部分木交換交叉においては，選択された部分木の意味が似ているかどうかは考慮していないため，大域探索や局所探索といった探索特性を制御することはできない．近年，この問題を解決するため，意味に基づく交叉が研究されている．前説で述べたように部分木の意味は，各入力に対する出力値を要素に持つベクトルとして定義される．よって選択された部分木同士の非類似度は，その部分木の意味ベクトル間の距離によって表現できる．その距離が大きければ大きいほど，部分木間の類似度は低いことになる．もし類似度の低い部分木を交換すれば，親個体のプログラムの振る舞いに非常に大きな影響を与える．結果として，生成される子個体のプログラムは，親個体のプログラムとはかなり異なるものとなる．これは大域的な探索に相当する．逆に，もし類似度の高い部分木同士を交換すれば，生成される子個体もプログラムの振る舞いの観点では，親個体と似たものになる．これは親個体周辺の局所探索とみなすことができる．

意味に基づく交叉として，Uy らは，Semantic Aware Crossover (SAC)[6] を提案した．SAC では，親個体と意味的に同一な子個体を生成しないように，選択された部分木の意味的な同一性をチェックしている．Semantic Similarity-based Crossover (SSC)[7] は SAC を拡張した手法である．局所探索を実行するため，意味的類似度の概念が導入された．親個体からの部分木の選択の際，意味的に類似しているが同一でない部分木が選ばれるまで，選択操作を繰り返す．これらの手法は，trial-and-error によるアプローチであり，また大域探索，局所探索の制御が困難である．

異なるアプローチとして，Hara らは，Semantic Control Crossover (SCC)[8]，および Rank-based SCC[9] を提案した．SCC では，交換される部分木が意味的類似度に基づいて確率的に選択される．加えて，探索特性を世代が進むにつれて変化させる．探索序盤は大域探索が実行され，世代が進むにつれて次第に局所探索へと移行する．一方，Rand-based SCC では，各世代において個体群を適応度によりランク付けし，高ランクの個体は局所探索となるように，低ランクの個体は大域探索となるように，類似度に基づいて交換対象の部分木を選択する．このように親個体の各ペアに対して，それらのランクに基づいて適切な探索特性を持つ交叉が割り当てられる．

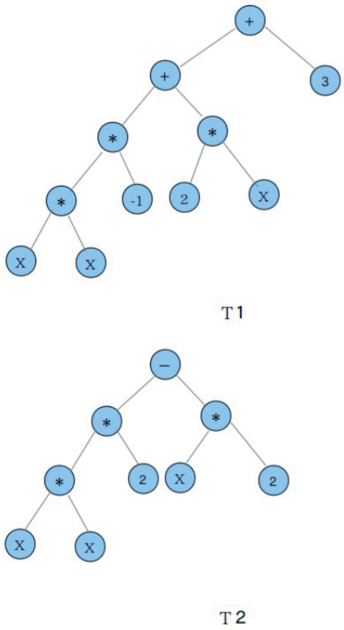


Fig. 3: Two parental individuals T_1 and T_2.

2.3 Geometric Semantic Genetic Programming

Geometric Semantic Genetic Programming (GSGP)[2, 3] は，Moraglio らにより提案された．GSGP における交叉では，子個体は部分木の交換によってではなく，親個体の凸結合によって生成される．2 つの親個体 T_1 と T_2 が与えられたとき，子個体 T_o は次のように生成される．

$$T_o = (T_1 * R) + (T_2 * (1 - R)), \tag{1}$$

ここで R は区間 $[0, 1]$ 内でランダムに生成された定数である．

例として，Fig. 3 に示すような，$T_1 = -X^2 + 2X + 3$, $T_2 = 2X^2 - 2X$ が 2 つの親個体となる場合を考える．変数 X への入力として，$X = 1, 2, 3, 4$ の 4 つのケースがある場合，T_1 と T_2 の意味は各々$(4, 3, 0, -5)$，$(0, 4, 12, 24)$ となる．もし，$R = 0.5$ とすると，Geometric Semantic Crossover により，Fig. 4 に示すような子個体 T_o が生成される．T_o の意味は $(2, 3.5, 6, 9.5)$ となる．子個体 T_o は，Fig. 5 に示すように，親個体 T_1 と T_2 の内分点に生成されたことになる．

GSGP における突然変異では，変異対象の個体 T_1 が与えられたとき，子個体 T_o は 2 つのランダムに作成した木構造プログラム TR_1 と TR_2 の差分を用いて，以下のように生成される．

$$T_o = T_1 + ms * (TR_1 - TR_2), \tag{2}$$

ここで，ms は変異幅 (mutation step) を表すパラメータである．ms を小さく取ると，T_o の意味は T_1 の意味に近くなる．言い換えれば，子個体は，意味空間において親個体の近傍に生成される．

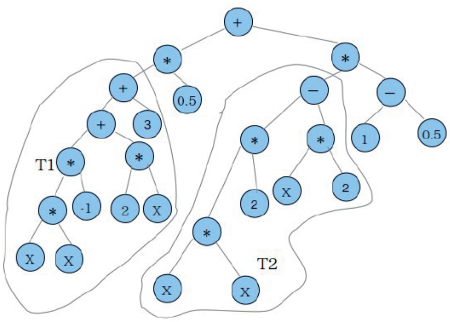


Fig. 4: An offspring created by the Geometric Semantic Crossover in $R = 0.5$.

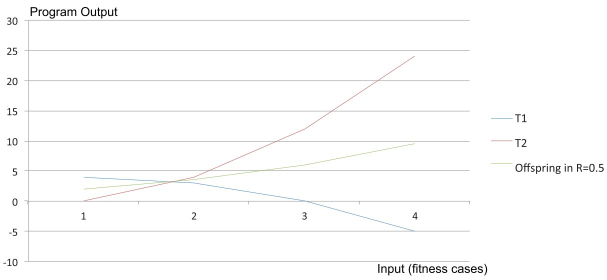


Fig. 5: An illustration of parents (T_1 and T_2) and an offspring created by conventional Geometric Semantic Crossover in the case of $R = 0.5$.

GSGP では，初期個体群は，標準的な GP と同様に，関数記号と終端記号をランダムに組み合わせて生成される．

3 提案手法

GSGP の概念図を Fig. 6 に示す．ここで，T_1 と T_2 は親個体が表す関数の出力を結んだ線を，P は同定対象のシステムの応答としてあらかじめ与えられる出力を結んだ線を表している．従来の GSGP では，2 つの親が張る空間の内部に子個体 T_o が生成される．この図から分かるように，生成される子個体は適応度が悪い個体（P に対してより遠い側にある個体）よりも適応度が改善するという特徴を持つ．しかし，この図のように P が親個体が張る空間の外側にある場合，両方の親個体よりも高い適応度を持つ（P により近くなる）子個体を生成することはできないという問題がある．この問題に対して，Hara ら [10] は，(1) 式の R の区間を拡張し，内分点だけでなく，外分点となる位置にも子個体を生成できるように改良した．しかし，R の値は乱数に従って決めており，必ずしも適切な位置に子個体が生成できるとは限らない．

これに対して，本論文では P があらかじめ与えられていることを利用して，P と T_1，T_2 との意味空間上の位置関係から適切な R の値を解析的に求めて利用する．具体的な方法を以下に示す．

P, T_1, T_2 の意味は，各々 D 次元ベクトルで表される（D は fitness case の数）．よって，D 次元の意味空間上で，P, T_1, T_2 の意味は，各々点で表現される．簡単のため，$D = 2$ とすると，P, T_1, T_2 の位置関係は，Fig. 7 の (a), (b), (c) の 3 通りに大別できる．Fig. 7(a) において，2 つの親個体 T_1, T_2 の凸結合により生成できる最も P に近い点は，点 P から線分 T_1T_2 に垂線を降ろした交点 P' である．このとき，R は以下の式で求

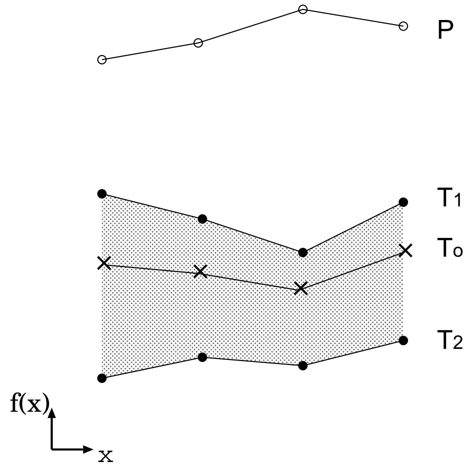


Fig. 6: Characteristics and problems of the conventional GSGP.

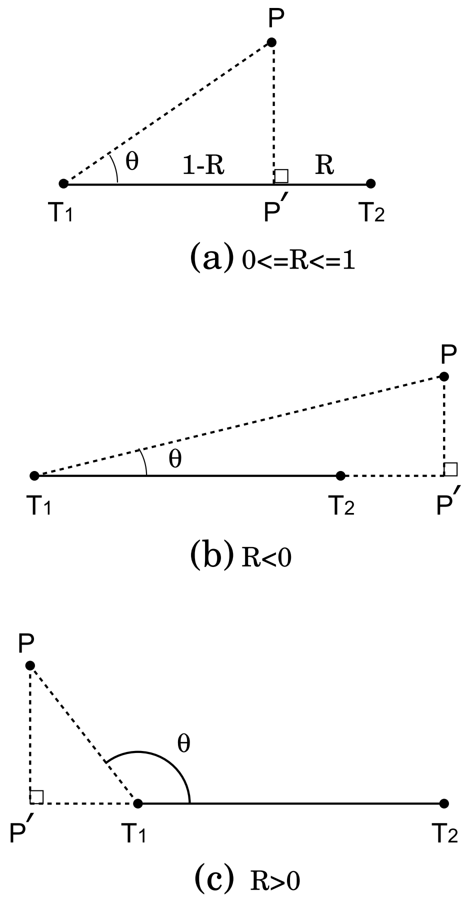


Fig. 7: Relationship of parents T_1, T_2 and target P in semantic space.

められる．

$$R = \frac{|T_1T_2| - |T_1P|cos\theta}{|T_1T_2|} \tag{3}$$

ここで，$\theta = \angle PT_1T_2$ である．同様に，Fig. 7(b)(c) のケースについても，2 つの親個体 T_1, T_2 のアフィン結合により生成できる点の中で P' が最も適応度の高い点となり，Fig. 7(a) のケースと同じ (3) 式で計算できる．ただし，Fig. 7(b) のケースは $R < 0$，Fig. 7(c) のケースは $R > 1$ となり，いずれも従来の GSGP では生成できないことが分かる．

Table 1: Parameter settings.

Population Size	100
Max Generations	2000
Tournament Size	4
Elite Size	1
Crossover Rate	0.8
Mutation Rate	0.2
Number of Trials	10

提案手法では，適応度に基づくトーナメント選択により選択された個体群の各ペアに対して，(3) 式により決定的に最適な R を求めた上で，その値を用いて (1) 式の Geometric Semantic Crossover を適用する．

4 実験と考察

4.1 実験環境およびパラメータ設定

ここでは，提案手法と従来の GSGP の性能比較を行う．Table 1 に，両手法に共通のパラメータ設定を示す．

対象は関数同定問題であり，与えられた入出力関係を満たす関数を見つけることが目標である．目的関数は，$f_1 = X^3 + X^2 + X$，$f_2 = 11X^2 + 13X + 17$，$f_3 = \sin(X^2)\cos(X) - 1$ の 3 種類で実験を行った．木構造プログラムの構成要素となる終端記号には $\{1.0, X\}$，関数記号には $\{+, -, *\}$ を用いた．ただし，交叉の際には，R を表す乱数定数も終端記号として利用される．各個体の適応度は，目的関数と個体が表す関数との 2 乗誤差の和とする．fitness case は，$[-1.0, 1.0]$ の区間で等間隔に 21 点をサンプリングした値 ($X = -1.0, -0.9, \ldots, 1.0$) を用いた．

4.2 実験結果と考察

目的関数 f_1, f_2, f_3 における従来の GSGP と提案手法の最良適応度の 10 試行平均の推移を Fig. 8, Fig. 9, Fig. 10 に各々示す．また，最終世代の最良適応度とその標準偏差を Table 2 に示す．これらの結果から，いずれの目的関数に対しても，提案手法が従来手法よりも良い成績をあげていることが分かる．

目的関数の違いによる性能への影響をみると，目的関数 f_2 に対して最も従来手法と提案手法の性能の差が大きい．関数 f_2 は係数の値を大きく設定しているため，ターゲットとなる意味ベクトルの各次元の値が他の問題よりも大きな値をとる．この結果，意味空間における初期個体集団の分布に対して，ターゲットがその外側に存在する可能性が高く，交叉では親個体の内分点に相当する個体しか作成できない従来の GSGP では最適化が困難であったと考えられる．これに対して，提案手法は親個体に近づくような外分点に相当する個体を作成できるため，効率的に探索が行えている．

5 おわりに

本論文では，関数同定問題を効率的に解くために，ターゲットの意味を利用した新たな Geometric Semantic Crossover を提案した．従来の GSGP との性能比較実験を行い，探索能力が大幅に改善できることを確認した．

今後は，探索の過程で個体群の分布が世代とともにどのように変化するか確認し，提案手法の探索の特性

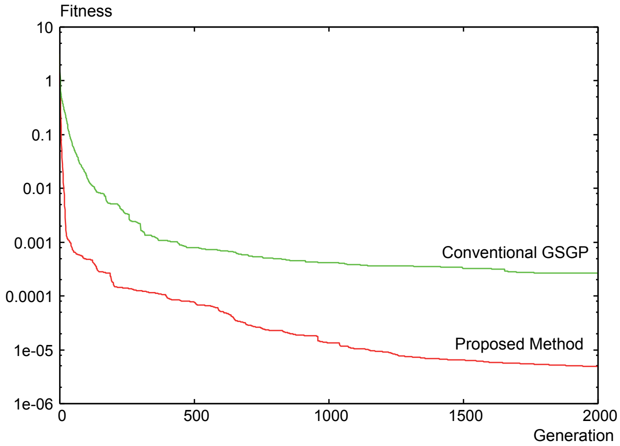


Fig. 8: Best fitness curves in f_1 problem.

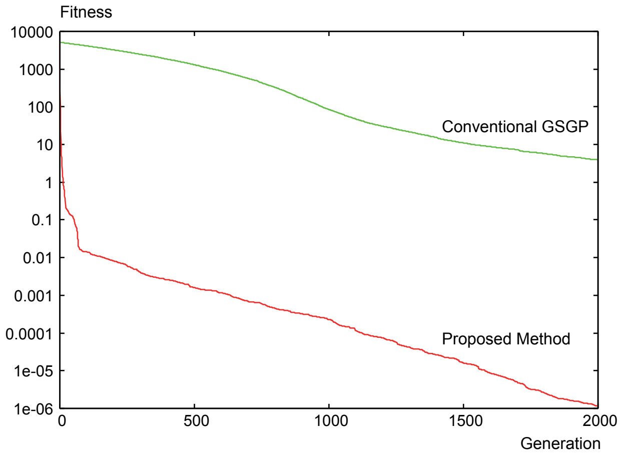


Fig. 9: Best fitness curves in f_2 problem.

をより詳細に分析する必要がある．また，提案した交叉方法を活かすためには，親個体のペアの作り方も重要である．本手法により適した選択方法についても設計を行いたい．

参考文献

1) John R. Koza: *Genetic Programming – On the Programming of Computers by Means of Natural Selection*, The MIT Press (1992)

2) Alberto Moraglio, Krzysztof Kwawiec, and Colin G. Jhonson: “Geometric Semantic Genetic Programming”, Parallel Problem Solving From Nature, XII, Part I, 21/31 (2012)

3) Leonardo Vanneschi, Mauro Castelli, LucaManzoni, and Sara Silva: “A New Implementation of Geometric Semantic GP and Its Application to Problems in Pharmacokinetics”, EuroGP 2013, LNCS 7831, 205/216 (2013)

4) Nicholas Freitag McPhee, Brian Ohs, and Tyler Hutchison: “Enumerating building block semantics in genetic programming”, University of Minnesota Morris Working Paper Series, Volume 3 Number 1, 1/11 (2007)

5) Nicholas Freitag McPhee, Brian Ohs, and Tyler Hutchison: “Semantic Building Blocks in Genetic Programming”, EuroGP 2008, LNCS 4971, 134/143 (2008)

6) Quang Uy Nguyen, Xuan Hoai Nguyen, and Michael O’Neill: “Semantic Aware Crossover for Genetic Programming: The Case for Real-Valued Function Regression”, EuroGP 2009, LNCS 5481, 292/302 (2009)

7) Quang Uy Nguyen, Xuan Hoai Nguyen, Michael and O’Neill, Bob Mckay, Edgar Galván-López:

Table 2: Comparison of the best fitness values.

	Conventional GSGP		Proposed Method	
f_1	2.609672e-04	(3.353077e-04)	**4.849679e-06**	**(1.102337e-05)**
f_2	3.914529	(1.020245e+01)	**1.180089e-06**	**(1.528702e-06)**
f_3	6.314171e-04	(4.628630e-04)	**6.314171e-04**	**(1.973358e-08)**

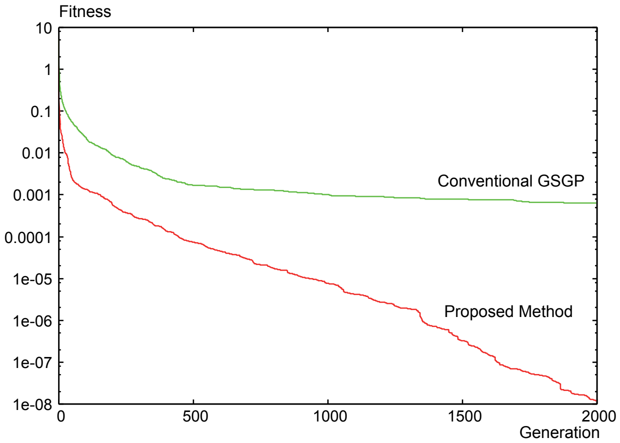


Fig. 10: Best fitness curves in f_3 problem.

"Semantically-based crossover in genetic programming: application to real-valued symbolic regression", Genetic Programming and Evolvable Machines, 12 (2), 91/119 (2011)

8) Akira Hara, Yoshimasa Ueno, and Tetsuyuki Takahama: "New crossover operator based on semantic distance between subtrees in Genetic Programming", 2012 IEEE International Conference on Systems, Man, and Cybernetics, 721/726 (2012)

9) Akira Hara, Jun-ichi Kushida, Takeyuki Nobuta, and Tetsuyuki Takahama: "Rank-based Semantic Control Crossover in Genetic Programming", 2014 IEEE International Conference on Systems, Man, and Cybernetics, 515/520 (2014)

10) Akira Hara, Jun-ichi Kushida, Kei Kisaka and Tetsuyuki Takahama: "Geometric Semantic Genetic Programming Using External Division of Parents", 2015 IIAI 4th International Congress on Advanced Applied Informatics, 189/194 (2015)

部分解評価を用いた対話型進化計算における定量的有効性評価

松本怜　染谷博司（東海大学）　加島智子（近畿大学）　折登由希子（広島大学）

Interactive Evolutionary Computation using Partial Solution and Quantification Approach for Performance Evaluation

∗R. Matsumoto, H. Someya (Tokai University), T. Kashima (Kinki University), Y. Orito (Hiroshima University)

Abstract– We propose an evaluation technique that focuses on partial solutions in interactive evoltionary computation (IEC) The partial solution evaluation is utilized in crossover and mutation operations. We applied the proposed IEC method to the menu planning problem. In experiment, we use characteristic score based on quantification approach.

Key Words: Interactive evolutionary computation, Menu planning problem, Combinatorial optimization

1　はじめに

遺伝的アルゴリズム（Genetic Algorithm）は，生物の進化淘汰に着想を得た最適化手法で，進化計算の一種である．情報・通信ネットワークや生産・工業分野への応用研究が盛んである．対話型進化計算（Interactive Evolutionary Computation）は，この遺伝的アルゴリズムの評価計算部分を，人間に任せる手法である．そうすることで，遺伝的アルゴリズムでは評価関数の設計が難しい・不可能であった問題でも最適化することができる．また，人間要素を取り込むことができるため，人間の知識や経験，感性に基づいた最適化が可能である．CG アートや音楽，人間らしいロボット制御，人間的知識が必要なデータマイニングなどに応用されている．また逆に，対話型進化計算を利用し最適化した結果から，人間の特性を調査する研究もあり，感性工学，人間科学分野への貢献も期待されている．このように幅広い分野で応用されている [1)]．

ところで，対話型進化計算の課題として，最適化時の人間への負荷が問題とされている．個体評価の際に，システムに何かしらの入力操作をしなければならない．遺伝的アルゴリズムの特性上，個体評価を繰り返すという単調作業になりやすいため，ユーザの心理的負担がある．このため，ユーザ負荷軽減のための効率化が求められている．具体的対策としては入出力インターフェースの改善や人間の評価の学習や推測を行うもの．進化計算自体の高速化などがあげられる．

遺伝的アルゴリズムに部分解を活用し，最適化する手法にウイルス進化説を踏襲した進化計算がある [2) 3)]．しかし，対話型進化計算に適用した例はない．

本研究では，解が部分解で構成される問題を対象に，部分解評価によって，評価次の負担を軽減する手法を提案する．ユーザの嗜好抽出は，より優れた解を集団内に生存させることで，実現する．本手法を献立計画問題に適用し，実装したシステムで，評価実験を行った．結果を示し，本手法の有用性を示す．

献立計画問題（Menu Planning Problem）は，最適な複数の料理の組合せを決定する最適化問題である．最適な食事は，個人により違い，多種多様に存在する．また，栄養素量はもちろん，自身の年齢や運動量（カロリーの考慮）・食べ合わせ（健康性・文化性・季節性の考慮）・費やせる調理時間・金銭的なコスト・以前食べた料理との相性（献立のマンネリ・飽きの考慮），食材の制限（アレルギー物質の考慮）などを考慮する必要があり，様々なものが，制約条件になる．そのため，万人が満足する一意の献立を作成することは困難であり，これらの制約条件を考慮しつつ，朝昼晩の献立を毎日考えるのは，各家庭や配食センターの栄養士，飲食店経営者などにとって，労力の大きい作業となる．

本研究は，提案する手法を献立計画問題へ当てはめ，本手法を実装したシステムによる被験者実験により，個人の嗜好を抽出し，これらの労力の軽減を試みる．なお，簡素化のため，本システムは，カロリーと個人の嗜好の 2 つを考慮する．

対話型進化計算の性能向上などについての報告は，被験者に対するアンケートを実施し，手法の評価を行う．報告の多くが被験者実験の結果に基づいて議論されている．しかし，このような評価方法は人によって評価が異なる恐れがあり，あまり好ましくない．本研究では，アンケート結果によらない評価指標を定義し，指標に基づいた手法の比較を行い，本手法の有効性を述べる．

2　提案手法

本研究で提案する，部分解を用いた，対話型進化計算の最適化手法を述べる．ある程度選別した個体群を，ユーザに提示し，対比較してもらうことで解評価をつけてもらう．さらに，評価された個体についてのみ，部分解にも評価をつけてもらう．部分解評価は，評価された部分解が，壊されないようにするために使用する．また，ユーザが選んだ個体を，必ず親にさせるため，固定的な親選択方式を用いる．

2.1　世代交代方法

世代交代の方法について述べる．まず，ランダムに初期集団を生成する．なお，本手法の集団サイズ $popsize$ は，n を自然数のパラメータとして，以下の様な離散値をとる．

$$popsize = ({}_nC_2) \times 2 \tag{1}$$

以下にユーザがどのように個体を評価するかを述べる．また，フローチャートを図 1 に示す．

(1) 初期集団の生成

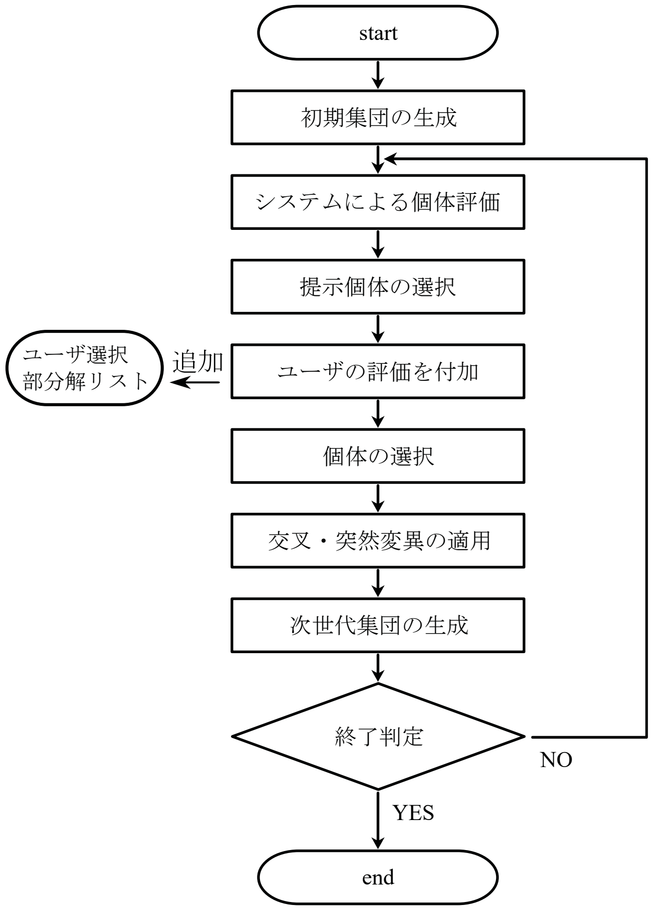


Fig. 1: 提案手法のフローチャート

- 集団サイズ $popsize$ の初期集団をランダムに生成する.

(2) 対話的評価と集団からの抽出

- システム側で個体の評価をし，トーナメントサイズ N_t のトーナメント選択を q 回実行し，q 個体を選ぶ．この q 個体をユーザに提示し，優れた個体を 1 つ選んでもらう．この最適だった 1 個体に対して，特にどの部分解が優れているか，1 つ選んでもらう (q 対比較)．これを N_i 回繰り返し，計 $q \times N_i$ 個体を抽出する．また，後述する定量的指標を算出するために，選ばれた部分解をユーザ選択部分解リスト L に追加しておく.

(3) 通常の抽出

- トーナメントサイズ N_t のトーナメント選択を $n - N_i$ 回実行し，集団から抽出する．最終的に対話的抽出と通常の抽出で計 n 個体を抽出する.

(4) 交叉・突然変異の適用

- 手順 (3) で選択した n 個体の中から 2 つ選ぶ全ての組合せで，それぞれ子を 2 個体ずつ作る．結果， $popzize$ 個体の子が作成される.

(5) 終了判定

- 手順 (2)〜手順 (4) を打ち切り世代数まで繰り返す.

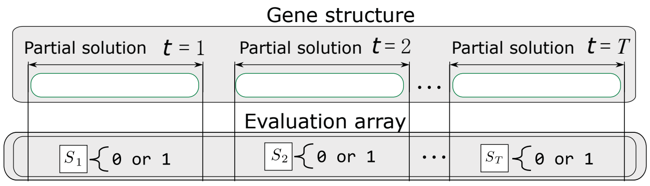


Fig. 2: Gene Structure

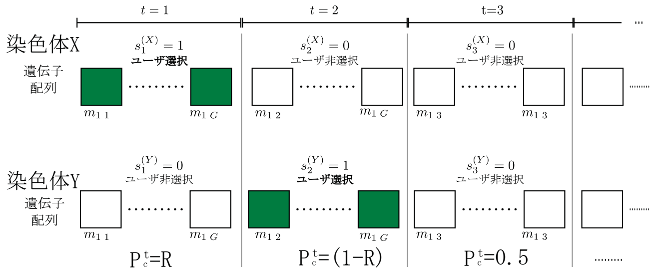


Fig. 3: 交叉の例

2.2 遺伝子表現

本手法の遺伝子表現方法を述べる．個体に対応する染色体 H は，表現型をコード化した遺伝子型配列 M と，部分解への評価結果を格納する評価配列 S から構成される（図 2）．M は, T 個の部分解 $N_t(t = 1, \ldots, T)$ から構成され，各 N_t は，G 個の遺伝子 $m_{t,g}(g = 1, \ldots, G)$ から構成される．S は，各 N_t に対応する評価変数 S_t により構成される．この評価情報は，バイナリ変数であり，以下のように保存する.

$$S_t = \begin{cases} 0 & \text{（ユーザが選択しなかった部分解の場合）} \\ 1 & \text{（ユーザが選択した部分解の場合）} \end{cases} \tag{2}$$

2.3 交叉率・突然変異

交叉は，一様交叉を行う．ただし，ユーザが評価した部分解を，壊さないように，部分解区間ごとに交叉・突然変異確率を決定する．個体 X の部分解 $S_t^{(X)}$ と個体 Y の部分解 $S_t^{(Y)}$ を交叉させる時，部分解 t の区間の交叉確率 P_t^C は，以下のようになる.

$$P_t^C = \begin{cases} 0.5 & (S_{b,t}^{(x)}, S_{b,t}^{(Y)} \text{が両方 0 の時}) \\ \left\{ \left(\overline{S_{b,t}^{(X)} \bigoplus S_{b,t}^{(Y)}} \right) \times \left((1-R) \times S_{b,t}^{(X)} + R \times S_{b,t}^{(Y)} \right) \right\} & \text{（それ以外）} \end{cases} \tag{3}$$

例えば，ユーザが染色体 X に対して $t = 1$，染色体 Y に対して $t = 2$ を選択した場合，$S_1^{(X)} = 1$，$S_2^{(Y)} = 1$ となり，交叉確率は図 3 のようになる．部分解 t の区間の突然変異確率 P_t^M は以下のようになる,

$$P_t^M = \begin{cases} R^t & (S_{b,t} = 0) \\ R^f & (S_{b,t} = 1) \end{cases} \tag{4}$$

例えば，ユーザが染色体 Z に対して $t = 2$ を選択した場合，

突然変異確率は図 4 のようになる.

3 実験方法と結果

本手法を 1 日 3 食（朝昼晩），1 日間の献立計画問題へ適用した．料理を和食に絞った 200 品目のデータベースを使用する．なお，遺伝子はこのデータベース

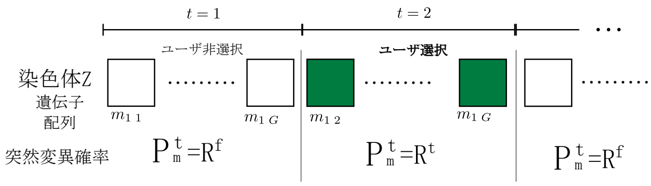


Fig. 4: 突然変異の例

の ID とする．本手法を実装したシステムにより実際にユーザに献立を作成してもらい，評価実験を行う．

3.1　献立生成システムの実装

本システムの評価関数は，個体が表す献立の 1 日の総カロリーを評価する関数と，q 対比較での比較で選択された個体には，良い評価を下す関数で構成されている．

3.2　実験手法

部分解評価をするもの（提案手法，Partial）と，部分解評価を使わない通常の対話型進化計算（w/o partial），ランダムに部分解評価を行う手法（Random）の 3 種類の手法を比較する．

3.3　手法の評価方法と評価指標

各世代の集団と，ユーザが選択した個体群を比較し，ユーザの嗜好が，集団内に反映されているか調査するため，以下の 3 つの指標を用意する．

Preference score は，各世代の集団と，ユーザが選択した個体群がどの程度似ているかを示す指標である．g 世代において，集団 P の各染色体に含まれる，遺伝子型配列内の IDi の数 $P_i^{(g)}$（ID の度数分布）と，ユーザ選択部分解リスト L 内の IDi の数 $L_i^{(g)}$（ID の度数分布）をかけ合わせたものである．Preference score は，以下の式で表せる．

$$\sum_{i}^{ID\ Max} P_i^{(g)} \times L_i^{(g)} \tag{5}$$

KL は，カルバック・ライブラー情報量に基づいた指標で，g 世代の集団 P の各染色体に含まれる，遺伝子型配列内の IDi の出現確率分布をモデル，ユーザ選択部分解リスト L 内の IDi の出現確率分布を真の分布とみなし，計算した数値である．

その被験者が，健康志向（カロリーバランスに優れた料理をよく好む『健康志向』なのか，そうではない『不健康志向』なのか，結果を比較するため，各世代の，集団内 ID のエントロピーの推移を調べる．各世代のエントロピー計算式は以下の式で表せる．

$$\sum_{i}^{idMax} -\left(\frac{P_i^{(g)}}{popsize \times T}\right) \times \log\left(\frac{P_i^{(g)}}{popsize \times T}\right) \tag{6}$$

標準的なユーザの数値を知るため，ユーザを模した個体評価モデルを作成し，シミュレーションを行ったこのモデルは，すべての料理にランダムに好み指数が設定されていて，固体内の料理を，この好み指数合計値が最大となる個体・部分解を，自動的に選択する．各被験者の結果を，標準的なユーザであるモデルと照らしあわせ，調査する．

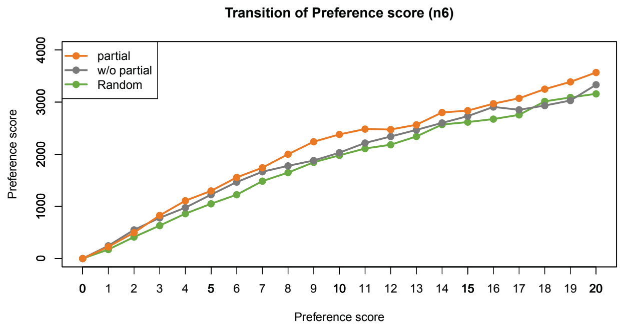


Fig. 5: Preference score の推移

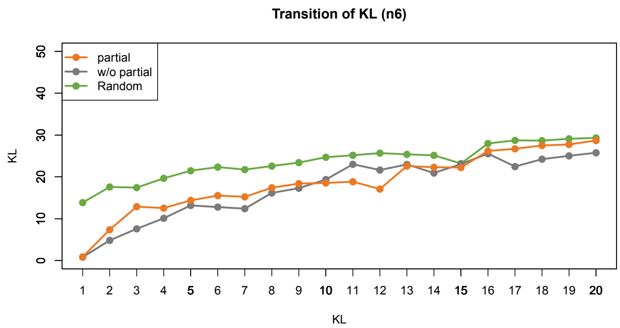


Fig. 6: KL 情報量の推移

3.4　実験結果と考察

各種法それぞれ 8 名の被験者に実験を行ってもらった．図 5 に各種法の被験者の Preference score 平均を示す．図 6 に各種法の被験者の KL 情報量平均を示す．図 7 に各種法の被験者のエントロピー平均を示す．なお，エントロピーの図 7 の agent 線に 39 回シミュレーションを行った平均値を示す．

Preference score は，Partial の手法が良く，本手法の有効性が示唆されている．しかし，KL 情報量は w/o partial の手法が良く，また，どの手法も，世代が進むごとに悪くなる傾向がある．これは，以前好きだった料理が，ユーザ選択部分解リスト内に残っている場合，ユーザ選択部分解リストと，集団の ID ヒストグラムの分布を見る限りでは，その分布に類似性がないためであると考えられる．単純に出現頻度の掛け算である Preference score がより適切にユーザの嗜好の変化に影響されずに，手法の良さを表せている．図 8 に世代の途中で，ユーザの嗜好が変化した時の，ユーザ選択部分解リストと，集団の ID ヒストグラムのイメージ図を示した．

4　まとめ

対話型進化計算は，評価計算部分を，人間に任せる手法である．評価関数の設計が難しい・不可能であった問題でも最適化することができ，人間要素を取り込むことができるが，ユーザに対しての負担が大きい問題がある．また，多くの報告では，得られた解が，ユーザの望むものだったかは，事後に行われるアンケートにたより，定性的な有効性評価をしている．本研究では，部分解を活用し，ユーザの負担軽減を試みた．また，本手法の有効性評価を行うため，定量的な指標を定義し，指標に基づいた評価を行った．指標による結果を踏まえ，世代を進める途中で，ユーザの好みが変

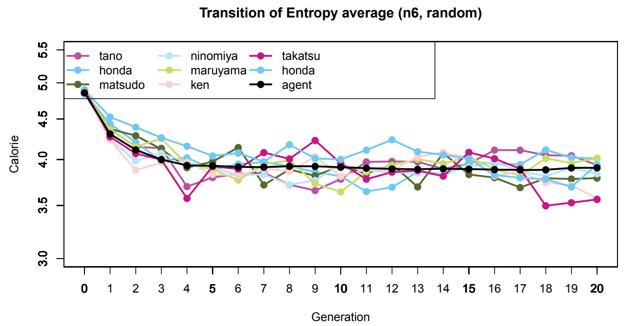


Fig. 7: エントロピー平均値の推移

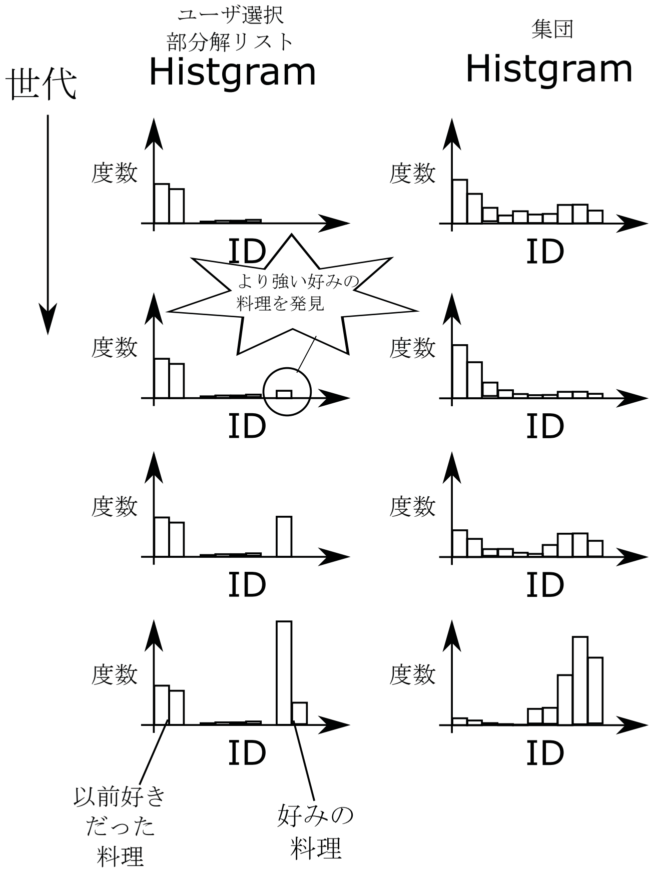


Fig. 8: ユーザ選択部分解リストと集団から見えるユーザの嗜好の変化

化してしまった場合の，指標の有効性の検討を行った．

謝辞: 実験の際に被験者を引き受けてくださった皆様に深く感謝いたします．また，料理データを提供してくださった Eat Smart,Inc. に感謝申し上げます．

参考文献

1) 電気学会 進化技術応用調査専門委員会：進化技術ハンドブック 第 1 巻 基礎編，114/122，近代科学社，(2010）

2) 中谷直司, 近藤邦雄：ウイルス進化論に基づく新しい遺伝的アルゴリズム，情報処理学会研究報告 (MPS), 数理モデル化と問題解決研究報告，1998-44，31/36 （1998）

3) 田村謙次：ウイルス進化型遺伝的アルゴリズムにおける感染手法による個体進化の相違に関する一考察，知能と情報：日本知能情報ファジィ学会誌：journal of Japan Society for Fuzzy Theory and Intelligent Informatics，20-5，791/799 （2008）

論理和によるファジィ推論モデルの提案とその性質

○関宏理（大阪大学）

Proposal of new fuzzy inference models using OR operation and its properties

∗Hirosato Seki (Osaka University)

Abstract– This paper firstly propose a fuzzy inference model by using OR operation. Moreover, an OR operation type fuzzy inference model with weights are also proposed. The proposed models are equivalent to inference results of minimum decision rule based on the rough set theory. Finally, the properties of the proposed models are shown in this paper.

Key Words: Approximate Reasoning, Fuzzy inference model, Rough set, OR operation, Equivalence

1 はじめに

Mamdani がファジィ推論の概念をスチームエンジン実験装置の制御へ適用して以来，様々な分野でファジィ推論の研究と応用が行われてきた [1)].

Mamdani のファジィ推論モデル [1)] や代数積–加算–重心モデルなどの IF-THEN 型ファジィ推論モデルでは，前件部に AND 演算を用いているため，1 つでも非発火となる場合は，その規則に対する適合度は 0 となってしまう．

一方，決定に最小限必要な属性や条件の長さが極小な決定ルール（以後，極小決定ルール）を決定表から抽出する方法として，ラフ集合に基づく方法がある [2, 3)]．矛盾を含んだデータを扱う場合，極小決定ルールを使用することにより矛盾を取り除くことが可能であることから，ファジィルールによる極小決定ルールの表現を可能にすることは非常に重要である．しかしながら，極小決定ルール以外のルールは使用しないため，通常のファジィ推論モデルにおけるファジィルールを使用する際には，非発火現象と同様の現象が起こる．このことから，一般にはすべてのファジィルールを用いた場合のファジィ推論では極小決定ルールを表現した推論結果が得られない．

したがって，本稿では前件部に論理和の概念を用いた OR 演算型ファジィ推論モデルを提案することにより，極小決定ルールと等価な推論結果が得られることを示す．また，OR 演算型ファジィ推論モデルに重みづけをした重み付き OR 演算型ファジィ推論モデルも提案する．本モデルも同様に，重みを 0 とすることにより極小決定ルールと等価な推論結果が得られることを示す．さらに，重みを [0,1] で与えたとき，極小決定ルール以外のルール，すなわち，極小決定ルールを導出する際に削減されたルールを抑制して使用することができるため，従来の極小決定ルールでは得られない推論結果が得られることについても言及する．

2 準備

2.1 ファジィ推論モデル

本章では代表的なファジィ推論モデルである Min–Max–重心モデル（Mamdani の推論モデル）[1)] と代数積–加算–重心モデル [4, 5, 6, 7)] について述べる．

一般的なファジィ推論モデルの規則は以下のような IF-THEN ルールで構成される．

$$\text{Rule } R_i = \left\{ \begin{array}{l} x_1 \text{ is } A_i^1 \text{ and } x_2 \text{ is } A_i^2 \text{ and } \cdots \text{ and } x_n \text{ is } A_i^n \\ \longrightarrow y \text{ is } B_i \end{array} \right. \tag{1}$$

ここで，$x_1, x_2, \ldots, x_n$ は前件部の入力変数，$A_i^1, A_i^2, \ldots, A_i^n$ はファジィ集合，B_i は後件部ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．なお，後件部のファジィ集合 B_i が関数に置き換えられた場合を T–S 推論モデル [8)]，定数に置き換えられた場合を簡略化ファジィ推論モデル [9)] と呼ぶ．

2.1.1 Min–Max–重心モデル

本節ではまず，Mamdani の推論モデルとして知られる Min–Max–重心モデル [1)] について説明する．Min–Max–重心モデルの規則は式 (1) のように与えられる．

入力 “$x_1^0, x_2^0, \ldots, x_n^0$” とファジィルール “$A_i^1, A_i^2, \ldots, A_i^n \longrightarrow B_i$” から導かれた各々の推論結果 B_i' は以下のように与えられる（図 1 参照）．

前件部 “$A_i^1, A_i^2, \ldots, A_i^n$” へ入力 “$x_1^0, x_2^0, \ldots, x_n^0$” が与えられたとき，その適合度 h_i は

$$h_i = \min\{A_i^1(x_1^0), A_i^2(x_2^0), \ldots, A_i^n(x_n^0)\} \tag{2}$$

のように与えられる．したがって，その規則の推論結

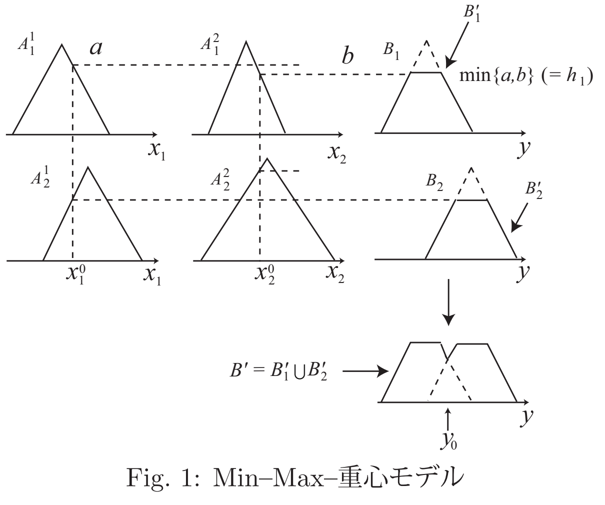


Fig. 1: Min–Max–重心モデル

果 B_i' は

$$B_i'(y) = \min\{h_i, B_i(y)\} \tag{3}$$

のように導くことができる．

式 (1) の最終推論結果 B' は B_1'，B_2'，…，B_M' を演算 max で結合することにより得ることができる．すなわち，

$$B'(y) = \max\{B_1'(y), B_2'(y), \ldots, B_M'(y)\} \tag{4}$$

となる．

推論結果 B' に対する代表値 y_0 は以下のように B' の重心を求めることにより得られる．

$$y_0 = \frac{\displaystyle\int y \cdot B'(y)dy}{\displaystyle\int B'(y)dy} \tag{5}$$

2.1.2 代数積–加算–重心モデル

次に本節では，代数積–加算–重心モデル [4, 5, 6, 7] について述べる（図 2 参照）．代数積–加算–重心モデルの規則は Min–Max–重心モデルと同様に，式 (1) で与えられる．

入力 "$x_1^0, x_2^0, \ldots, x_n^0$" とファジィルール "$A_i^1, A_i^2, \ldots, A_i^n \longrightarrow B_i$" から得られた各推論結果 B_i' は以下のように求められる．

前件部 "$A_i^1, A_i^2, \ldots, A_i^n$" へ入力 "$x_1^0, x_2^0, \ldots, x_n^0$" が与えられたとき，その適合度 h_i は以下のように求められる．

$$h_i = A_i^1(x_1^0) \cdot A_i^2(x_2^0) \cdots A_i^n(x_n^0) \tag{6}$$

ここで "·" は代数積を意味する．したがって，その推論結果 B_i' は以下のように導かれる．

$$\begin{aligned} B_i'(y) &= A_i^1(x_1^0) \cdots A_i^n(x_n^0) \cdot B_i(y) \\ &= h_i \cdot B_i(y) \end{aligned} \tag{7}$$

式 (1) の最終推論結果 B' は B_1'，B_2'，…，B_M' を代数和 (+) で統合することにより得ることができる．すなわち，

$$B'(y) = B_1'(y) + B_2'(y) + \cdots + B_M'(y) \tag{8}$$

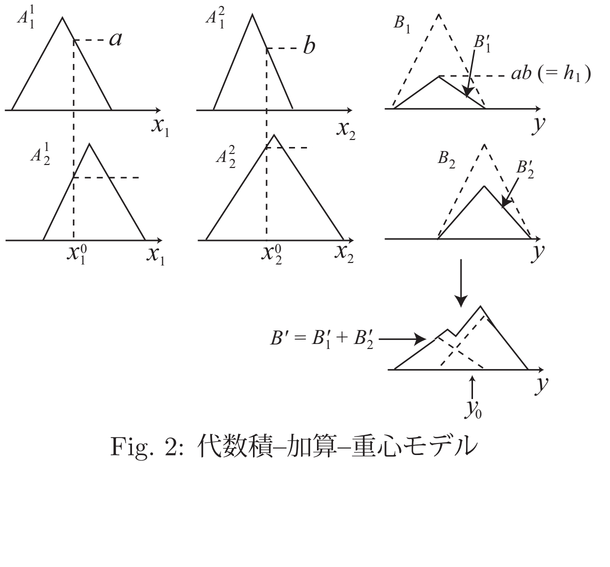


Fig. 2: 代数積–加算–重心モデル

となる．

推論結果 B' の代表値 y_0 は重心法により以下で求めることができる．

$$y_0 = \frac{\displaystyle\int y \cdot B'(y)dy}{\displaystyle\int B'(y)dy} \tag{9}$$

2.2 ラフ集合 [2, 3]

2.2.1 近似

議論の対象となる全体集合を U，U 上の同値関係を R とする．U の R による同値類 $[x]_R$ は，同値関係 R のもとで互いに区別できない対象を表わす．部分集合 $X \subseteq U$ に対して，R による X の**下近似**$R_*(X)$ 及び**上近似**$R^*(X)$ が以下のように定義される．

$$R_*(X) = \{x \in U \mid [x]_R \subseteq X\} \tag{10}$$
$$R^*(X) = \{x \in U \mid [x]_R \cap X \neq \emptyset\} \tag{11}$$

下近似と上近似の対 $(R_*(X), R^*(X))$ を X の**ラフ集合**と呼ぶ．X のラフ集合は，集合の包含関係の意味で，X の上下方向からの近似を表わす．$R_*(X)$ は X への帰属が疑わしくない対象の集合と解釈され，$R^*(X)$ は X に帰属する可能性のある対象の集合と解釈される．

2.3 ラフ集合と決定表

決定表は $(U, C \cup D, V, \rho)$ の 4 項組で表される．ここで，U は対象の集合，C は条件属性集合，D は決定属性集合である．V は属性値の集合，ρ は $U \times C\ or\ D$ から V の中への写像を表す．表 1 に決定表の例を示す．この表は 6 人の患者 (Patient) の集合を頭痛 (Headache：以後 H)，筋肉痛 (Musclepain：以後 M)，体温 (Temperature：以後 T) から流感 (Flu：以後 F) か否かを類別したデータである．$U = \{x_1, x_2, ..., x_6\}$，$C = \{\mathrm{H,M,T}\}$，$D = \{\mathrm{F}\}$，$V = \{\text{yes,no,normal,high,very high}\}$ である．対象の集合が p 個の決定クラス D_k $(k = 1, 2, \cdots, p)$ に分割されたとすると，条件属性の部分集合 $A \subseteq C$ に基づき，各決定クラスの下近似，上近似は次のように定義される．

$$A_*(D_k) = \{x \in U \mid [x]_A \subseteq D_k\} \tag{12}$$
$$A^*(D_k) = \{x \in U \mid [x]_A \cap D_k \neq \emptyset\} \tag{13}$$

ただし，$[x]_A$ は A による対象 x の同値類を表わす．

表 1 において，流感である患者の集合 $D_1 = \{x_1, x_2, x_3, x_4\}$ と流感でない患者の集合 $D_2 = \{x_5, x_6\}$ とする．また，$A = \{\mathrm{H,M,T}\}$ とすると

Table 1: 決定表の例

Patient	Headache	Musclepain	Temperature	Flu
x_1	no	yes	high	yes
x_2	yes	no	high	yes
x_3	yes	yes	very high	yes
x_4	no	yes	very high	yes
x_5	no	yes	normal	no
x_6	yes	no	high	no

$A_*(D_1) = \{x_1, x_3, x_4\}$，$A^*(D_1) = \{x_1, x_2, x_3, x_4, x_6\}$，$A_*(D_2) = \{x_5\}$，$A^*(D_2) = \{x_2, x_5, x_6\}$ が得られる．

決定表に関する指標として次式に示す**近似精度**$\alpha_A(D_k)$ と**近似の質**$\gamma_A(D_k)$ が提案されている [10]．

$$\alpha_A(D_k) = \frac{|A_*(D_k)|}{|A^*(D_k)|} \tag{14}$$

$$\gamma_A(D_k) = \frac{|A_*(D_k)|}{|D_k|} \tag{15}$$

$|X|$ は集合 X の要素数である．$\alpha_A(D_k)$ は属性集合 A により決定クラス D_k がどの程度近似できるかを示し，$\gamma_A(D_k)$ は属性集合 A の情報により決定クラス D_k がどの程度の要素が明確に判定できるかを示している．

表 1 において, $D_1 = \{x_1, x_2, x_3, x_4\}$, $D_2 = \{x_5, x_6\}$, $A = \{\text{H,M,T}\}$ とすると，$\alpha_A(D_1) = 3/5 = 0.6$, $\gamma_A(D_1) = 3/4 = 0.75$, $\alpha_A(D_2) = 1/3 \cong 0.33$, $\gamma_A(D_2) = 1/2 = 0.5$ となる．

2.3.1 決定表と決定ルール

決定表は，条件属性の値に対する決定属性の値を示す決定ルールを与えている．例えば，表 1 の第 2 行は

$$[\text{H} = \text{yes}] \wedge [\text{M} = \text{no}] \wedge [\text{T} = \text{high}] \Rightarrow [\text{F} = \text{yes}]$$

なるルール，すなわち,「頭痛があり，筋肉痛がなく，体温が高ければ，流感である」という決定ルールを示している．

決定ルール $\Delta \Rightarrow \Gamma$ の質を評価するために，次のような指標が考えられている．

$$\text{Cer}(\Gamma \mid \Delta) = \frac{|\Gamma \wedge \Delta|}{|\Delta|} \tag{16}$$

$$\text{Cov}(\Gamma \mid \Delta) = \frac{|\Gamma \wedge \Delta|}{|\Gamma|} \tag{17}$$

$$\text{Supp}(\Gamma \mid \Delta) = \frac{|\Gamma \wedge \Delta|}{|U|} \tag{18}$$

$\text{Cer}(\Gamma \mid \Delta)$ は**確信度**と呼ばれ，そのルールがどの程度正しいかを示している．$\text{Cov}(\Gamma \mid \Delta)$ は**被覆度**と呼ばれ，そのルールにより結論 Γ を説明できる対象の割合を示している．$\text{Supp}(\Gamma \mid \Delta)$ は**支持度**と呼ばれ，そのルールを満たす対象が全体の何割にあたるかを示している．

表 1 において，$\Delta = [\text{H} = \text{no}] \wedge [\text{M} = \text{yes}]$，$\Gamma = [\text{F} = \text{yes}]$ とすると，

$$\text{Cer}(\Gamma \mid \Delta) = \frac{|x_1, x_4|}{|x_1, x_4, x_5|} = \frac{2}{3} \cong 0.67$$

$$\text{Cov}(\Gamma \mid \Delta) = \frac{|x_1, x_4|}{|x_1, x_2, x_3, x_4|} = \frac{2}{4} = 0.5$$

$$\text{Supp}(\Gamma \mid \Delta) = \frac{|x_1, x_4|}{|x_1, x_2, x_3, x_4, x_5, x_6|} = \frac{2}{6} \cong 0.33$$

となる．

2.3.2 ラフ集合による知識獲得

ここでは，決定表に内在する極小決定ルールをすべて導出する方法である決定行列について述べる．決定表 $(U, C \cup D, V, \rho)$ が与えられたとき，決定属性集合 $B \subseteq D$ の属性値に基づき，対象の集合が p 個の決定クラス $D_k(k = 1, 2, ..., p)$ に分割されたとする．このとき，決定クラス D_k に応じて**決定行列**の (i, j) 成分は式 (19) のように定義される．

$$M_{ij}^k = \{(a, \rho(x_i, a)) \mid \rho(x_i, a) \neq \rho(x_j, a)\}$$
$$i \in K_k^+ = \{i \mid x_i \in C_*(D_k)\},\ j \in K_k^- = \{j \mid x_j \notin D_k\} \tag{19}$$

ここで，$\mathcal{L}(M_{ij}^k)$ を (i, j) 成分の要素の論理和とすると，式 (20) で示す論理式の主加法標準形を求めれば，各連言項が $x \in D_k$ を導く条件の長さが極小な決定ルールの条件部となる．

$$\bigvee_{i \in K_k^+} \bigwedge_{j \in K_k^-} \mathcal{L}(M_{ij}^k) \tag{20}$$

このような操作をすべての決定クラス D_k について行えば，決定表内に内在する極小決定ルールのすべてが求められる．

表 1 において，$D_1 = \{x_1, x_2, x_3, x_4\}$ とした場合，決定クラス D_1 の決定行列は以下のようになる．

	x_5	x_6
x_1	{(T,high)}	{(H,no),(M,yes)}
x_3	{(H,yes),(T,very high)}	{(M,yes),(T,very high)}
x_4	{(T,very high)}	{(H,no),(M,yes),(T,very high)}

式 (20) より

$$\bigvee_{i \in K_k^+} \bigwedge_{j \in K_k^-} \mathcal{L}(M_{ij}^k)$$
$$\begin{aligned}
=&[\text{T} = \text{high}] \wedge ([\text{H} = \text{no}] \vee [\text{M} = \text{yes}]) \\
&\vee ([\text{H} = \text{yes}] \vee [\text{T} = \text{very high}]) \wedge ([\text{M} = \text{yes}] \vee \\
&\qquad [\text{T} = \text{very high}]) \\
&\vee [\text{T} = \text{very high}] \\
=&([\text{H} = \text{no}] \wedge [\text{T} = \text{high}]) \vee ([\text{M} = \text{yes}] \wedge [\text{T} = \text{high}]) \\
&\vee ([\text{H} = \text{yes}] \wedge [\text{M} = \text{yes}]) \vee [\text{T} = \text{very high}]
\end{aligned}$$

これにより，次に示す $r1, r2, r3, r4$ の 4 つの決定ルールが得られる．

$$r_1 : [\text{H} = \text{no}] \wedge [\text{T} = \text{high}] \Rightarrow [\text{F} = \text{yes}]$$
$$r_2 : [\text{M} = \text{yes}] \wedge [\text{T} = \text{high}] \Rightarrow [\text{F} = \text{yes}]$$
$$r_3 : [\text{H} = \text{yes}] \wedge [\text{M} = \text{yes}] \Rightarrow [\text{F} = \text{yes}]$$
$$r_4 : [\text{T} = \text{very high}] \Rightarrow [\text{F} = \text{yes}]$$

3 論理和を考慮したファジィ推論モデルの提案

たとえば，表 1 のような決定表をファジィ推論へ適用することを考える．このとき，表 1 を基にしたルールを使用する場合，Patient x_2 と x_6 のように，x_2 と x_6 の条件属性は Headache is no, Musclepain is yes, Tempareture is high と同じであるのに対し，決定属性は異なる．このような矛盾を含んだデータを用いた場合，十分な精度が得られない場合が多々存在する．このような矛盾を含んだデータに対して，**2.3.2** で示すようなラフ集合による極小決定ルールを用いることにより，矛盾を含むことなく精度の良い推論結果を得られ

ることが知られている[2, 3]．しかしながら，一般的なファジィ推論を用いた場合，ファジィルールの前件部はANDで結ばれているため，1つの入力でも非発火となるならば，その規則の適合度は0になってしまう．したがって，一般的なファジィルールからすべてのルールを用いた場合，極小決定ルールのような欠損ルールのような推論結果は得られないことがわかる．

このことから，本稿では欠損ルールに対応し，1つ以上非発火現象が起きたとしても推論結果が得られるために，AND部分をOR演算に置き換えた論理和を考慮したファジィ推論モデル（以後，OR演算型ファジィ推論モデル）の提案を行う．OR演算型ファジィ推論モデルのファジィルールは以下のように与えられる．

$$\text{Rule } R_i = \left\{ \begin{array}{l} x_1 \text{ is } A_i^1 \text{ or } x_2 \text{ is } A_i^2 \text{ or } \cdots \text{ or } x_n \text{ is } A_i^n \\ \longrightarrow y \text{ is } B_i \end{array} \right. \tag{21}$$

ここで，$x_1, x_2, \ldots, x_n$ は前件部の入力変数，$A_i^1, A_i^2, \ldots, A_i^n$ はファジィ集合，B_i は後件部ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．

入力“$x_1^0, x_2^0, \ldots, x_n^0$”とファジィルール“$A_i^1, A_i^2, \ldots, A_i^n \longrightarrow B_i$”から得られたOR演算型ファジィ推論モデルの各推論結果 B_i' は以下のように求められる．

前件部“$A_i^1, A_i^2, \ldots, A_i^n$”へ入力“$x_1^0, x_2^0, \ldots, x_n^0$”が与えられたとき，ORで結合されていることから，その適合度 h_i は以下のように求められる．

$$h_i = A_i^1(x_1^0) + A_i^2(x_2^0) + \cdots + A_i^n(x_n^0) \tag{22}$$

ここで“+”は代数和を意味する．したがって，その推論結果 B_i' は以下のように導かれる．

$$\begin{aligned} B_i'(y) &= \{A_i^1(x_1^0) + \cdots + A_i^n(x_n^0)\} * B_i(y) \\ &= h_i * B_i(y) \end{aligned} \tag{23}$$

ここで，演算 $*$ は代数積やmin演算などの t-normを意味する．

式(26)の最終推論結果 B' は B_1'，B_2'，…，B_M' をmax演算などの t-conorm，あるいは代数和(+)で統合することにより得ることができる．すなわち，

$$B'(y) = B_1'(y) * B_2'(y) * \cdots * B_M'(y) \tag{24}$$

となる．ここで，演算 $*$ は t-conorm，あるいは代数和を意味する．

推論結果 B' の代表値 y_0 はmin–max–重心モデルや代数積–加算–重心モデルと同様に以下で求めることができる．

$$y_0 = \frac{\displaystyle\int y \cdot B'(y)dy}{\displaystyle\int B'(y)dy} \tag{25}$$

一例として，図3に代数和–Max–重心モデルの場合の推論図を示す．

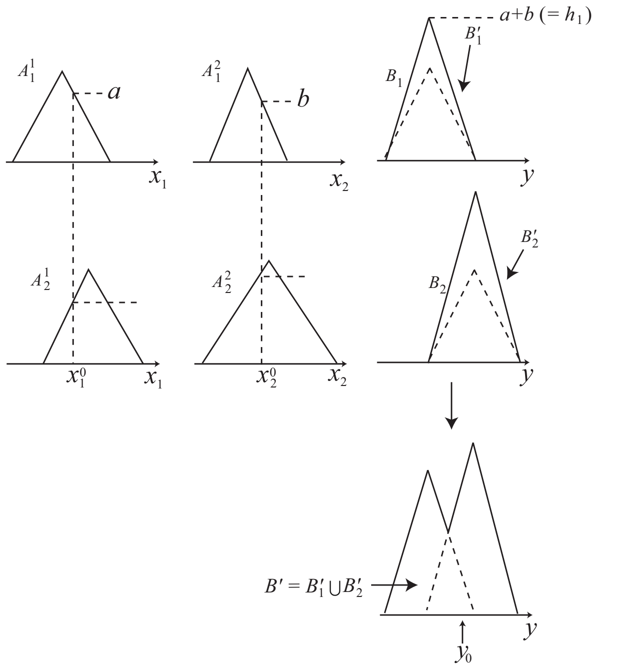


Fig. 3: 代数和–Max–重心モデルの場足のOR演算型ファジィ推論モデル

4 OR演算型ファジィ推論モデルの性質

本章ではファジィ推論の等価性の観点からOR演算型ファジィ推論モデルの特徴を述べる．

著者らは以下のようにファジィ推論の等価性を定義している．

定義[11, 12]：任意の入力が与えられたとき，2つのファジィ推論モデルの推論結果が等しいとする．このとき，2つのファジィ推論モデルは等価であると呼ぶ．

従来のAND型ファジィ推論では前件部がAND演算を用いるため，1つでも非発火状態になった場合にはその規則が0になってしまう．したがって，等価性を考えたとき，ラフ集合のような極小決定ルールのような欠損ルールと等価な推論結果は得られない．

一方，OR演算型ファジィ推論では，前件部同士の演算は式(22)のように加算を用いるため，入力とルールの間で非発火現象が起こったとしても，それ以外の部分を加算することにより欠損ルールと同等の結果を得ることができる．

また，OR演算型ファジィ推論モデルが非発火現象以外で欠損ルールと等価な推論結果が得られることを考えた場合，前件部に重みを付加し，欠損部分については重みを0とすることが考えられる．すなわち，前件部に対して重み付けを行うOR演算型ファジィ推論モデル（以後，重み付きOR演算型ファジィ推論モデル）は以下のように拡張することができる．

$$\text{Rule } R_i = \left\{ \begin{array}{l} x_1 \text{ is } w_i^1/A_i^1 \text{ or } x_2 \text{ is } w_i^2/A_i^2 \text{ or } \cdots \text{ or } x_n \text{ is } w_i^n/A_i^n \\ \longrightarrow y \text{ is } B_i \end{array} \right. \tag{26}$$

ここで，w_i^1/A_i^1 の / は通常の割り算ではなく，セパレータを表し，w_i^1 は A_i^1 の重要度を意味する．

たとえば，ラフ集合による極小決定ルールの導出の際に使用しない条件属性が i 番目のルールにおける j

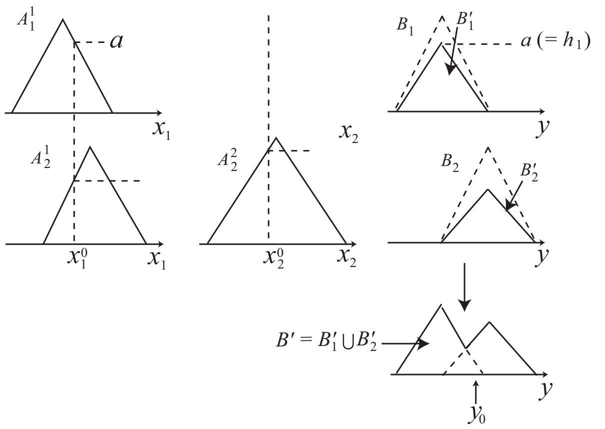


Fig. 4: 極小決定ルールを用いた場合

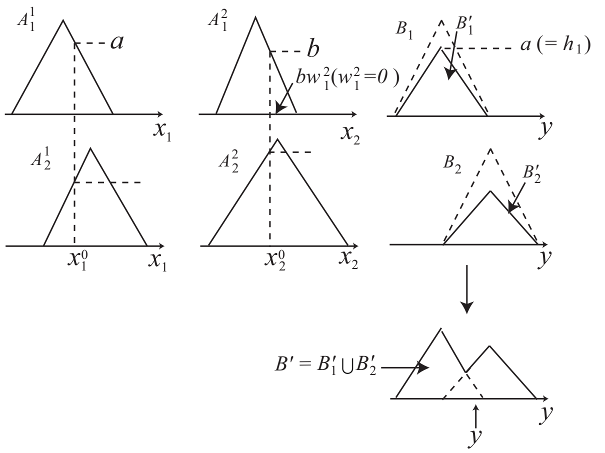


Fig. 5: $w_1^2 = 0$ の場合の重み付き OR 演算型ファジィ推論モデル

番目の入力に対する条件属性であるとする．このとき，$w_i^j = 0$ とすることにより，重み付き OR 演算型ファジィ推論モデルはラフ集合による極小決定ルールと等価な推論結果が得られる（図 4,5 参照）．

ラフ集合による極小決定ルールを表現する際には欠損部分に対応する重みを 0 とすることにより等価な推論結果が得られるが，$w_i^j \in [0,1]$ とすることにより，極小決定ルールでは得ることのできない推論結果が得られる．極小決定ルールを使用する場合，矛盾を取り除くことができる特徴がある一方で，取り除いた部分が本当に必要がないのかという問題があることが知られている．しかしながら，本提案モデルでは，OR 演算と重み付けにより推論結果を得ることから，たとえば，欠損部分の重み w_i^j を 0 ではなく 0.2 と設定したとすると，抑制してそのルールを使用するという表現が可能になる．

重み付き OR 型ファジィ推論モデルでは極小決定ルールよりもパラメータ数が増加するものの，推論結果の統合に加算や Max 演算を用いる．著者らは，統合に加算を用いた場合，重心法は以下の式に書き直すことができることを示している [11]．

$$
\begin{aligned}
y_0 &= \frac{\int yB'(y)dy}{\int B'(y)dy} \\
&= \frac{\sum_{i=1}^{M} S_i' y_i}{\sum_{i=1}^{M} S_i'} \\
&= \frac{\sum_{i=1}^{M} h_i S_i y_i}{\sum_{i=1}^{M} h_i S_i}
\end{aligned} \tag{27}
$$

ここで S_i は後件部ファジィ集合の面積，y_i はその重心，S_i' は i 番目の規則の推論結果の面積を表し，$i = 1, 2, \ldots, M$，M は規則の総数である．

同様に，著者らは Max 演算で統合する場合，重心法による推論結果は以下のように書き換えることができることも示している [13]．

$$
\begin{aligned}
y_0 &= \frac{\int yB'(y)dy}{\int B'(y)dy} \\
&= \frac{\sum_{i=1}^{M} S_i' y_i - \sum_{i=1}^{M-1} S_i'^{o} y_i^{o}}{\sum_{i=1}^{M} S_i' - \sum_{i=1}^{M-1} S_i'^{o}}
\end{aligned} \tag{28}
$$

ここで $S_i'^{o}$ 各規則からの推論結果の Max 統合における重複部分の面積を，y_i^o はその重心を表す．

したがってパラメータ数は増加するものの，統合に加算や Max 演算を用いることができることから，上記による高速演算を使用することが可能である．

また，上述ではファジィルールの前件部は OR で結合されたもののみを議論したが，AND と OR を混合したものとして以下のようなファジィルールを考えることも可能である．

$$
\begin{aligned}
&\text{Rule } R_i = \\
&\begin{cases} x_1 \text{ is } A_i^1 * x_2 \text{ is } A_i^2 * \cdots * x_n \text{ is } A_i^n \\ \longrightarrow y \text{ is } B_i \end{cases}
\end{aligned} \tag{29}
$$

ここで，$*$ は and または or を意味する．$x_1, x_2, \ldots, x_n$ は前件部の入力変数，$A_i^1, A_i^2, \ldots, A_i^n$ はファジィ集合，B_i は後件部ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．

この場合も，and で結合された部分の 1 つでも非発火である場合は，and 部分はすべて適合度が 0 になるものの，or 演算が含まれている場合はその部分が加算として使用されるため，その規則の適合度は 0 にならないという特徴を持つ．

5 おわりに

一般的に，従来のファジィ推論モデルでは極小決定ルールに基づく推論結果と等価な推論結果を得ることはできない．しかしながら，本稿では OR 演算を用い

たOR演算型ファジィ推論モデルを提案し，本モデルは極小決定ルールと等価な推論結果が得られることを示した．また，OR演算型ファジィ推論モデルの前件部に重みづけしたモデルを提案し，重み付きOR演算型ファジィ推論モデルも同様に極小決定ルールと等価な推論結果が得られることを示した．さらに，重みを0から1の間をとることにより，極小決定ルールで使用しない部分を抑制して使用することにより，極小決定ルールでは得られない推論結果を得ることができることをも示した．極小決定ルールと等価な推論結果が得られるということは，矛盾を含んだデータに対しても精度良く対応するため，医療診断などへの応用が期待される．

参考文献

1) E. H. Mamdani: Application of fuzzy algorithms for control of simple dynamic plant, Proc. IEE, **121** 1585/1588 (1974).

2) Z.Pawlak: Rough sets, International Journal of Information Computer Science, **11**-5, 341/356 (1982).

3) 水口, 水本：被覆度を考慮したラフ集合による知識獲得及びファジィ判別分析, バイオメディカル・ファジィ・システム学会大会講演論文集, 35/38 (2005)

4) M. Mizumoto: Fuzzy controls under various fuzzy reasoning methods, Information Sciences, **45**, 129/151 (1988).

5) M. Mizumoto: Fuzzy controls under product–sum–grvity method and new fuzzy control methods, Fuzzy Control Systems (ed. A. Kandel and G. Langholz), CRC Press, 275/294 (1993).

6) B. Kosko: Neural Networks and Fuzzy Systems, Prentice Hall (1992).

7) B. -G. Hu, G. K. I. Mann, and R. G. Gosine: A systematic study of fuzzy PID controllers—function-based evaluation approach, IEEE Transactions on Fuzzy Systems, **9**-5, 699/712 (2001).

8) T. Takagi and M. Sugeno: Fuzzy identification of systems and its Applications to modeling and control, IEEE Trans. Syst., Man, Cybern., **SMC-15**-1, 116/132 (1985).

9) H. Ichihashi: Iterative fuzzy modeling and a hierarchical network, Proc. 4th IFSA Congress of Engineering, 49/52 (1991).

10) 森, 田中, 井上：ラフ集合と感性, 海文堂, 163/181 (2004).

11) H. Seki and M. Mizumoto: On the equivalence of fuzzy inference methods—part 1: basic concept and definition, IEEE Trans. Fuzzy Systems, **19**-6, 1097/1106 (2011).

12) H. Seki, H. Ishii and M. Mizumoto: On the generalization of single input rule modules connected type fuzzy reasoning method, IEEE Trans. Fuzzy Systems, **16**-5, 1180/1187 (2008).

13) 関，水本：Mamdaniモデルの等価性とType-2ファジィ推論モデルの高速演算に関する一考察, 第28回ファジィシステムシンポジウム講演論文集, 888/893 (2012).

リズム現象における位相・周期感度の解析法

○森　禎弘　黒江康明　（京都工芸繊維大学）

Analysis Method of Phase and Period Sensitivities in Rhythmic Phenomena

∗Y. Mori and Y. Kuroe (Kyoto Institute of Technology)

Abstract– Sensitivity analysis is fundamental and essential in analysis and design in any system. This paper discusses a method of sensitivity analysis of rhythmic phenomena which are found in various systems such as physical systems, biological systems and human societies and so on. Sensitivity analysis of rhythmic phenomena is very difficult because rhythms appear autonomously as periodic phenomena in nonlinear systems and only few studies have been done. In this paper we propose an analysis method of phase and period sensitivities in rhythmic phenomena. We first give the definition of phase for periodic tra ectories of nonlinear systems and derive a strict expression of phase and period sensitivities by introducing Poincaré map. Based on the expression we derive an efficient computer algorithm to calculate phase and period sensitivities by intruducing sensitivity equations or ad oint equations. It is shown that the proposed analysis method makes it possible to obtain both the period and phase sensitivities accurately.

Key Words: Rhythm phenomenon, Period sensitivity, Phase sensitivity, Nonlinear system, Poincaré map, Sensitivity equation, Ad oint equation

1　はじめに

リズム現象は自然現象や物理現象，あるいは経済システム，社会システムなどありとあらゆるシステムで見られる非線形現象である．たとえば，生体内では歩行運動や呼吸，心臓の鼓動などさまざまなリズム現象が多く見られる．またサーカディアン・リズムとよばれる睡眠などの１日を周期とするリズム現象も存在する．これらのリズム現象は，生体内の特定のニューラルネットワークあるいは遺伝子ネットワークが司っていることが知られており，これらの現象の生理的および数理的な解明が盛んにされてきた．また工学の分野においてはこれらの結果に基づきリズム現象を工学的に実現し，応用する研究も行われるようになってきている．また生体だけでなく，リズム現象を数理的にモデリングしそのメカニズムを解明しようとする研究，それらの成果を利用しようとする研究，あるいはそれらの成果にヒントを得てシステムの解析，設計における種々の問題を解決しようとする研究がさまざまな分野で行われるようになってきている．たとえば，複数の異なるリズム現象が相互に影響しあうとやがて一つのリズム現象に同期する同期現象と呼ばれる現象がさまざまな分野で見られ，この現象の解明に関する研究が盛んに行われている．これらの成果を自律分散制御やシステム創発的設計などに応用する研究も行われるようになってきている．

一方，システムのパラメータが微小変化したとき，システムの振る舞い，特性や性能がどのように変化するかを調べること，すなわち感度解析は，システムの解析，設計におけるさまざまな場面で必要となり，あらゆるシステムにとって非常に重要な問題である．システムの振る舞い，特性や性能のパラメータに関する感度は，通常，それらのパラメータに関する微分として定義される．また，システムの性能や応答に対しある評価関数が与えられ，それを最小あるいは最大にするようなパラメータを求める最適化問題において，これらの問題を勾配法など種々開発されている最適化アルゴリズムを用いて解く際，評価関数のパラメータに関する微分値が必要となり，この場合も感度解析が必要となる．本稿の目的は，リズムを対象としてその感度解析法を開発することである．

システムの感度解析の研究は古くからなされ，これまで多くの研究がある．ところがリズム現象，すなわち周期現象に関する感度の研究はそれほど多くない．その理由は，システムにおける周期解は一般に初期状態には依存しない非線形現象の定常状態として自律的に決まり，そのパラメータの依存性の表現を得ることが困難なためである．周期現象の感度に関してこれまでいくつかの研究があるが [1, 2, 3]，近似的な扱いが含まれる，その具体的な計算法が示されていない，計算効率が考慮されていないというような問題がある．筆者らは既にリズム現象の感度解析として，特に位相と周期に対する感度解析法を提案し，精度が良くかつ計算効率の良い感度解析アルゴリズムを提案している [4, 5, 6]．このアルゴリズムでは，感度方程式を導いて，あるいは随伴方程式を導いて感度を求めることができる [6]．本稿では，随伴方程式を導いて感度を求める方法を示し，感度方程式を導いて求める場合との比較を行う．

リズム現象は多くの場合，非線形自律システムの周期状態としてモデル化される．そこで，本稿では非線形自律システムを対象として，まず，その周期状態の位相の定義を与える．そのうえで，定義した位相に関するパラメータ感度の表現を導出する．このような非線形現象の感度は，解析的に求めることがほとんど不可能で最終的には数値的に求めることになり，その際，計算精度および計算効率が問題となる．そこで本稿では，導いた感度の表現より，リズム現象における周期と位相のパラメータ感度を数値的に計算するための精度が良くかつ計算効率の良いアルゴリズムを示す．このとき，感度方程式を導いて求める方法と随伴方程式を導いて求める方法が考えられる．本稿では位相感度と周期感度を随伴方程式に基づいて求める方法を示し，そのための精度が良く計算効率の高い計算アルゴリズムを導く．そして，感度方程式を導いて求める方法との比較を行う．提案法は，リズム現象を生じさせる通常の安定なリミットサイクルの感度だけなく，たとえ

ばカオスアトラクターに埋め込まれた不安定なリミットサイクルの感度も求めることができる方法となっている．

2　対象とするシステムと数学的準備

本稿で取り扱うリズム現象は，次の微分方程式で表される非線形自律システムの周期解としてモデル化できるものを対象とする．

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{p}) \tag{1}$$

ここで，$\boldsymbol{x}(t)$ は状態ベクトル，$\boldsymbol{p}$ はパラメータベクトルであり，それぞれ n 次元ベクトル（$\boldsymbol{x} \in \mathbb{R}^n$），$p$ 次元ベクトル（$\boldsymbol{p} \in \mathbb{R}^p$）であるとする．$\boldsymbol{f}$ は非線形のベクトル関数である（$\boldsymbol{f} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}^n$）．$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p})$ は，$\boldsymbol{x}$ および $\boldsymbol{p}$ に関してそれぞれ 1 階連続微分可能とする．また，$\boldsymbol{\phi}(t, \boldsymbol{x}_0, \boldsymbol{p})$ を初期条件 $\boldsymbol{x}(0) = \boldsymbol{x}_0$ としたときの，(1) 式で表されるシステムの時刻 t での解とする．すなわち，$\boldsymbol{\phi}(t, \boldsymbol{x}_0, \boldsymbol{p})$ は次式を満たす．

$$\boldsymbol{\phi}(t, \boldsymbol{x}_0, \boldsymbol{p}) = \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}(\boldsymbol{\phi}(\tau, \boldsymbol{x}_0, \boldsymbol{p}), \boldsymbol{p}) d\tau \tag{2}$$

ここで，(1) 式のシステムは，非自明な孤立した周期解，すなわちリミットサイクルを持つと仮定し，これがリズム現象を実現しているものとする．この周期解の状態空間における軌道を $\boldsymbol{\gamma}$ とし，この軌道上の一点を $\boldsymbol{x}_\gamma$ とする．$\boldsymbol{\gamma}$ が周期軌道であるということより，任意の点 $\boldsymbol{x}_\gamma \in \boldsymbol{\gamma}$ に対し，

$$\boldsymbol{x}_\gamma = \boldsymbol{\phi}(T, \boldsymbol{x}_\gamma, \boldsymbol{p}) \tag{3}$$

が成立する．ここで，T は周期解の周期であり，周期 T はシステムのパラメータ $\boldsymbol{p}$ に依存して決まる．本稿の目的は，(1) 式で表されるシステムにおけるリズム現象，すなわち (1) 式のシステムの周期軌道 $\boldsymbol{\gamma}$ を対象として，その特徴的な量である位相と周期のパラメータ感度を求める方法を提案することである．

以下では，本稿で必要となるいくつかの数学的な準備を与える．システムの周期解の定性的な性質や安定性を解析するための重要なツールとしてポアンカレ写像 (Poincaré Map) が知られている [7, 8]．ポアンカレ写像の基本的な考え方は，連続時間システムの周期軌道の解析を，それより 1 次元低い離散時間システムに置き換えて解析することであり，次のように定義される．

【ポアンカレ写像の定義】

周期解の軌道を $\boldsymbol{\gamma}$ とする．$\boldsymbol{\gamma}$ 上の一点を $\boldsymbol{x}_{\gamma_0}$ とし，この点で $\boldsymbol{\gamma}$ と横断的に交わる超平面（曲面）Σ を考える．$U \subset \Sigma$ を点 $\boldsymbol{x}_{\gamma_0}$ の近傍とする．点 $\boldsymbol{x} \in U$ に対して

$$\boldsymbol{P}(\boldsymbol{x}) = \boldsymbol{\phi}(\mathcal{T}, \boldsymbol{x}, \boldsymbol{p}) \tag{4}$$

で定義される写像 $\boldsymbol{P} : U \to \Sigma$ をポアンカレ写像とよぶ．ただし $\mathcal{T}(\boldsymbol{x})$ は点 $\boldsymbol{x}$ から出発した解軌道 $\boldsymbol{\phi}(t, \boldsymbol{x}, \boldsymbol{p})$ が最初に Σ にもどってくる時刻である．すなわち，$\mathcal{T}(\boldsymbol{x}) = \min\{t | \boldsymbol{\phi}(t, \boldsymbol{x}, \boldsymbol{p}) \in \Sigma\}$ である．

明らかに，$\boldsymbol{x}_{\gamma_0} = \boldsymbol{P}(\boldsymbol{x}_{\gamma_0}), \mathcal{T}(\boldsymbol{x}_{\gamma_0}) = T$ が成立し，周期軌道上の点は，ポアンカレ写像の不動点となる．なお，$\boldsymbol{\gamma}$ と横断的に交わる超平面（曲面）Σ は，ポアンカレ断面と呼ばれる．

3　位相・周期感度の解析法

本稿の目的は，周期軌道 $\boldsymbol{\gamma}$ の位相や周期に対するパラメータ感度を求めるためにはまず，それらのパラメータ対する依存性の表現を得る必要がある．ところが，システムにおける周期解は初期状態には依存しない非線形現象の定常状態として自律的に決まり，その周期や位相についてパラメータの依存性の表現を得ることは一般に困難である．筆者らは先に，周期軌道の位相と周期に関するパラメータの依存性の数学的表現を導き，それを基にして位相感度と周期感度を計算する方法を提案した [4, 5, 6]．ここでは，この提案方法を説明する．

3.1　位相・周期感度の表現の導出

非線形システムの周期軌道の位相に対するパラメータに関する依存性を導出するためには，まずその位相をどのように定義するかを明らかにする必要がある．本稿では，(1) 式で表されるシステムの周期軌道 $\boldsymbol{\gamma}$ に対する位相を次のように定義する．$\boldsymbol{\gamma}$ の 1 周期内においてある注目する事象 (event) が起こる時刻を考える．この時刻を適切に定めた基準となる時刻からの差として定義し，これを t_{ev} と書くことにする．時刻 t_{ev} はシステムのパラメータ $\boldsymbol{p}$ に依存しているので，このことを明らかにするため $t_{ev}(\boldsymbol{p})$ と表す．そのように定義した事象 (event) が起こる時刻の位相 $\theta(\boldsymbol{p})$ を次のように定義する．

$$\theta(\boldsymbol{p}) = 2\pi \frac{t_{ev}(\boldsymbol{p})}{T(\boldsymbol{p})} \tag{5}$$

本稿では，このように定義した位相に関するパラメータ $\boldsymbol{p}$ に対する感度を計算する方法を議論する．パラメータ感度は一般に，そのパラメータに関する微分値として定義されるので，(5) 式の両辺を $\boldsymbol{p}$ で微分すると感度が次式のように得られる．

$$\frac{\partial}{\partial \boldsymbol{p}}\theta(\boldsymbol{p}) = 2\pi \frac{\frac{\partial t_{ev}(\boldsymbol{p})}{\partial \boldsymbol{p}} T(\boldsymbol{p}) - t_{ev}(\boldsymbol{p}) \frac{\partial T(\boldsymbol{p})}{\partial \boldsymbol{p}}}{T^2(\boldsymbol{p})} \tag{6}$$

従って位相感度は，事象が起こる時刻に対する感度 $\partial t_{ev}(\boldsymbol{p})/\partial \boldsymbol{p}$ および，周期に関する感度 $\partial T(\boldsymbol{p})/\partial \boldsymbol{p}$ を求めることができれば求まることになる．

一方，注目する事象としては，たとえば，注目する状態変数がある値になる，あるいは注目する状態変数が極大値，あるいは極小値をとるというようなことが考えられる．このような時刻は一般に，システムの状態 $\boldsymbol{x}$ とパラメータ $\boldsymbol{p}$ に依存する $\mathbb{R}^n \times \mathbb{R}^p \to R$ なる関数 $\mathcal{E}_v(\cdot)$ を用いて，次式を満足する時刻として表される．

$$\mathcal{E}_v(\boldsymbol{x}(t_{ev}), \boldsymbol{p}) = 0 \tag{7}$$

たとえば，状態 $\boldsymbol{x}$ の i 番目の要素が零になるという事象が起こる時刻を t_{ev} とすると

$$x_i(t_{ev}) = 0$$

と表すことができ，$\mathcal{E}_v$ は，$\mathcal{E}_v = x_i$ となる．また，状態 $\boldsymbol{x}$ の i 番目の要素が極値をとるという事象が起こる時刻を t_{ev} とすると

$$\frac{d}{dt}x_i(t_{ev}) = 0$$

と表すことができ，$\mathcal{E}_v$ は，$\mathcal{E}_v = \frac{d}{dt}x_i = f_i(\boldsymbol{x}, \boldsymbol{p})$ となる．以下では，(7) 式を用いて (6) 式で定義された位相感度を求める際必要となる $\partial t_{ev}(\boldsymbol{p})/\partial \boldsymbol{p}$ および，$\partial T(\boldsymbol{p})/\partial \boldsymbol{p}$ の表現を導く．

位相感度と周期感度の数学的に厳密な表現は，周期軌道の 1 点がポアンカレ写像の不動点になることを利用して導出できる．注目する事象の起こる時刻 t_{ev} は，対象とする周期解 $\boldsymbol{\gamma}$ がポアンカレ断面 Σ を横切る時刻を基準として定義することにし，その時刻を初期時刻 $t = 0$ とする．ポアンカレ断面は次式で表されるとする．

$$\sigma(\boldsymbol{x}) = 0 \tag{8}$$

ここで，σ は $\sigma : \mathbb{R}^n \to \mathbb{R}$ なる写像である．対象とする周期軌道 $\boldsymbol{\gamma}$ に対し，ポアンカレ断面は適切に設定できるとし，$\sigma(\boldsymbol{x})$ は既知であるとする．周期軌道とポアンカレ断面の交点 $\boldsymbol{x}_{\gamma_0}$ は (3) 式を満たすので，次式が成立する．

$$\boldsymbol{x}_{\gamma_0} = \boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}) \tag{9}$$

また $\boldsymbol{x}_{\gamma_0}$ は Σ 上の点であることから

$$\sigma(\boldsymbol{x}_{\gamma_0}) = 0 \tag{10}$$

が成立する．また，(7) 式の $\boldsymbol{x}(t)$ は周期解なので

$$\boldsymbol{x}(t) = \phi(t, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}) \tag{11}$$

と書け，このとき (7) 式は次の様に書くことができる．

$$\mathcal{E}_v(\phi(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}), \boldsymbol{p}) = 0 \tag{12}$$

ここで，パラメータ $\boldsymbol{p}$ が $\Delta\boldsymbol{p}$ だけ変化し $\boldsymbol{p} + \Delta\boldsymbol{p}$ となったとする．このとき周期軌道 $\boldsymbol{\gamma}$ は，周期状態を維持したままポアンカレ断面 Σ との交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T，および事象が起こる時刻 t_{ev} がそれぞれ

$$\begin{aligned} &\boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0} \\ &T + \Delta T \\ &t_{ev} + \Delta t_{ev} \end{aligned}$$

と変化したとする．すなわち次式が成立するとする．

$$\boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0} = \boldsymbol{\phi}(T + \Delta T, \boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0}, \boldsymbol{p} + \Delta\boldsymbol{p}) \tag{13}$$

また事象が起こる時刻が満たす (7) 式はこの変化により，次のようになる．

$$\mathcal{E}_v(\phi(t_{ev} + \Delta t_{ev}, \boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0}, \boldsymbol{p} + \Delta\boldsymbol{p}), \boldsymbol{p} + \Delta\boldsymbol{p}) = 0 \tag{14}$$

さらに $\boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0}$ は Σ 上の点なので (10) 式より次式が成立する．

$$\sigma(\boldsymbol{x}_{\gamma_0} + \Delta\boldsymbol{x}_{\gamma_0}) = 0 \tag{15}$$

ここで，(13) 式の右辺，(14) 式の左辺，および (15) 式の左辺をテーラー展開し，それぞれ (9) 式，(12) 式，および (10) 式を用いることにより得られる式において，$\Delta\boldsymbol{x}_{\gamma_0}$, ΔT と Δt_{ev} を未知数と考えて整理すると，最終的に次ページの (16) 式が得られる [6]．なお，(16) 式の左辺における $\mathcal{E}_v(\boldsymbol{x}(t_{ev}), \boldsymbol{p})$ はすべて $\mathcal{E}_v(\phi(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}), \boldsymbol{p})$ と書くべきであるが，スペースの関係でこのように略記している．以後もこのように略記する．

ここで記述の簡単のため，(16) 式の左辺の係数行列を $\boldsymbol{S}$，右辺第 1 項の係数行列を $\boldsymbol{V}$ として (16) 式を次のように表す．

$$\boldsymbol{S} \begin{bmatrix} \Delta\boldsymbol{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix} = \boldsymbol{V}\Delta\boldsymbol{p} + O(\Delta^2) \tag{17}$$

この方程式を解くことにより，次式が得られる．

$$\begin{bmatrix} \Delta\boldsymbol{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix} = \boldsymbol{S}^{-1}\boldsymbol{V}\Delta\boldsymbol{p} + O(\Delta^2) \tag{18}$$

これより感度が次の様に求まる．

$$\begin{aligned} \frac{\partial x_{\gamma_0 j}}{\partial p_i} &= \left\{\boldsymbol{S}^{-1}\boldsymbol{V} \text{ の } j \text{ 行 } i \text{ 列要素}\right\} \\ \frac{\partial T}{\partial p_i} &= \left\{\boldsymbol{S}^{-1}\boldsymbol{V} \text{ の } (n+1) \text{ 行 } i \text{ 列要素}\right\} \\ \frac{\partial t_{ev}}{\partial p_i} &= \left\{\boldsymbol{S}^{-1}\boldsymbol{V} \text{ の } (n+2) \text{ 行 } i \text{ 列要素}\right\} \end{aligned}$$

これが求めるべき感度の表現である．この表現に基づいてこれらの感度を求めるためには，行列 $\boldsymbol{S}$ および $\boldsymbol{V}$ の各要素を求める必要があり，これらをいかに精度よく，かつ効率よく計算するかが問題となる．次項ではこれらを計算するアルゴリズムを説明する．

3.2 感度計算アルゴリズム

3.2.1 周期解の求解法

前項で導いた (18) 式の行列 $\boldsymbol{S}$ および $\boldsymbol{V}$ を求めるためには，まず周期軌道 $\boldsymbol{\gamma}$ を求める必要がある．ここでは，計算効率と精度を考慮した周期解の求解法について説明する．

(9) 式と (10) 式は $\boldsymbol{x}_{\gamma_0}$ と T に関する代数方程式ととらえることができるので，$\boldsymbol{y} = [\boldsymbol{x}'_{\gamma_0}\ T]'$ とおき

$$\boldsymbol{F}(\boldsymbol{y}) = \boldsymbol{0}$$

なる代数方程式を考える．ただし

$$\boldsymbol{F}(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{x}_{\gamma_0} - \boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}) \\ \sigma(\boldsymbol{x}_{\gamma_0}) \end{bmatrix} \tag{19}$$

で，$\boldsymbol{A}'(\boldsymbol{x}')$ は行列 $\boldsymbol{A}$(ベクトル $\boldsymbol{x}$) の転置行列（ベクトル）である．この代数方程式をニュートン法で解くことにするとその反復計算式は次の様に表される．

$$\boldsymbol{y}^{i+1} = \boldsymbol{y}^i - D\boldsymbol{F}(\boldsymbol{y}^i)^{-1}\boldsymbol{F}(\boldsymbol{y}^i) \tag{20}$$

ただし $D\boldsymbol{F}(\boldsymbol{y})$ は $\boldsymbol{F}(\boldsymbol{y})$ の $\boldsymbol{y}$ に関するヤコビ行列で，次のように表される．

$$D\boldsymbol{F}(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{I} - \frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}) & -\frac{\partial}{\partial t}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p}) \\ \frac{\partial}{\partial \boldsymbol{x}}\sigma(\boldsymbol{x}_{\gamma_0}) & 0 \end{bmatrix} \tag{21}$$

(20) 式のニュートン法の反復計算において $\boldsymbol{F}(\boldsymbol{y}^i)$ および $D\boldsymbol{F}(\boldsymbol{y}^i)$ を適切に計算する必要がある．

$\boldsymbol{F}(\boldsymbol{y}^i)$ の第 1 項 $\boldsymbol{\phi}(T^i, \boldsymbol{x}^i_{\gamma_0}, \boldsymbol{p})$ は，初期条件を $\boldsymbol{x}(0) = \boldsymbol{x}^i_{\gamma_0}$ とし (1) 式を $t = 0$ から $t = T^i$ まで解くことによ

$$\begin{bmatrix} \boldsymbol{I}-\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}) & -\frac{\partial}{\partial t}\boldsymbol{\phi}(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}) & 0 \\ \frac{\partial}{\partial \boldsymbol{x}}\mathcal{E}_v(\boldsymbol{x}(t_{ev}),\boldsymbol{p})\frac{\partial}{\partial \boldsymbol{x}_0}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p}) & 0 & \frac{\partial}{\partial \boldsymbol{x}}\mathcal{E}_v(\boldsymbol{x}(t_{ev}),\boldsymbol{p})\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p}) \\ \frac{\partial}{\partial \boldsymbol{x}}\sigma(\boldsymbol{x}_{\gamma_0}) & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}) \\ -\frac{\partial}{\partial \boldsymbol{x}}\mathcal{E}_v(\boldsymbol{x}(t_{ev}),\boldsymbol{p})\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p})-\frac{\partial}{\partial \boldsymbol{p}}\mathcal{E}_v(\boldsymbol{x}(t_{ev}),\boldsymbol{p}) \\ 0 \end{bmatrix} \Delta \boldsymbol{p} + O(\Delta^2) \tag{16}$$

り得られる．第2項 $\sigma(\boldsymbol{x}^i_{\gamma_0})$ は，ポアンカレ断面の (8) 式は既知としているので計算できる．

ヤコビ行列 $D\boldsymbol{F}(\boldsymbol{y}^i)$ の (1,1) ブロックの中の $\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p})$ は，次のように計算できる．(2) 式の両辺を $\boldsymbol{x}_0$ で微分すると，

$$\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(t,\boldsymbol{x}_0,\boldsymbol{p}) = \int_0^t \frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{\phi}(\tau,\boldsymbol{x}_0,\boldsymbol{p}),\boldsymbol{p})\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(\tau,\boldsymbol{x}_0,\boldsymbol{p})d\tau \tag{22}$$

が得られる．$\boldsymbol{z}_{x_0}(t)=\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(t,\boldsymbol{x}_0,\boldsymbol{p})$ とおくと $\boldsymbol{z}_{x_0}(t)$ は次式を満たすことが容易にわかる．

$$\frac{d}{dt}\boldsymbol{z}_{x_0}(t)=\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{\phi}(t,\boldsymbol{x}_0,\boldsymbol{p}),\boldsymbol{p})\boldsymbol{z}_{x_0}(t), \quad \boldsymbol{z}_{x_0}(0)=\boldsymbol{I} \tag{23}$$

ここで，$\boldsymbol{I}$ は単位行列である．従って $\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p})$ は，感度方程式と呼ばれる (23) 式の微分方程式を，$t=0$ から $t=T^i$ まで解いたときの解 $\boldsymbol{z}_{x_0}(T^i)$ として求まる．

一方，随伴方程式を導入して求める方法は，以下のようになる [5]．(23) 式の感度方程式に対する随伴方程式は次式で与えられる．

$$\frac{d}{d\tau}\hat{\boldsymbol{z}}_{x_0}(\tau)=D\boldsymbol{f}'\cdot\hat{\boldsymbol{z}}_{x_0}(\tau), \quad \hat{\boldsymbol{z}}_{x_0}(0)=\boldsymbol{e}_j, \quad j=1,2,\cdots,n \tag{24}$$

ここで，$D\boldsymbol{f}=\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}),\boldsymbol{p})$，$\tau=T^i-t$ である．(24) 式を解いて得られる $\hat{\boldsymbol{z}}_{x_0}$ より，$\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p})$ は次式により求まる [5]．

$$\left\{\boldsymbol{z}_{x_{0i}}(T^i)\right\}_j=\left\{\hat{\boldsymbol{z}}_{x_0}(0)\right\}_i \tag{25}$$

ここで $\{\boldsymbol{y}_w\}_j$ はベクトル $\boldsymbol{y}_w$ の第 j 要素である．

またヤコビ行列 $D\boldsymbol{F}(\boldsymbol{y}^i)$ の (1,2) ブロックの $\frac{\partial}{\partial T}\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p})$ は, (2) 式において $t=T$ とした式の両辺を t で微分することにより

$$\frac{\partial}{\partial t}\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p})=\boldsymbol{f}(\boldsymbol{\phi}(T^i,\boldsymbol{x}^i_{\gamma_0},\boldsymbol{p}))$$

が得られ，この式の右辺を計算することにより求まる．

以上により (20) 式のニュートン法の反復計算を進めることができ，ニュートン法が収束すると，ポアンカレ断面と周期軌道の交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T，および周期軌道 $\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$, $0\le t\le T$ が求まることになる．

なお，以上で説明した周期解の求解法は，安定なリミットサイクルだけでなく，たとえばカオスに埋め込まれた不安定なリミットサイクルも求められる方法となっている [9]．

3.2.2 行列 $\boldsymbol{S}$ と $\boldsymbol{V}$ の要素の計算法

行列 $\boldsymbol{S}$ と $\boldsymbol{V}$ の各要素も，対象とするシステムに対して初期状態 $\boldsymbol{x}_0$ に対する感度方程式およびパラメータ $\boldsymbol{p}$ に対する感度方程式を導くことにより，あるいはそれらに対して随伴方程式を導くことによってその計算法を導出することができる [6]．ここでは，その一例として $\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\phi(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ を，随伴方程式を用いて計算する方法を説明する．まず事象が起こる時刻 t_{ev} を求める必要がある．これは次のようにして求める．すなわち，前項で説明したニュートン法により求まったポアンカレ断面と周期軌道の交点 $\boldsymbol{x}_{\gamma_0}$ を初期状態として (1) 式を解くことにより周期軌道 $\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ が求まるので，これが (7) 式を満足する時刻として事象が発生する時刻 t_{ev} が求まる．この時刻 t_{ev} と周期 T を用いて $\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\phi(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ は以下のように求めることができる．

まず，$\frac{\partial}{\partial \boldsymbol{p}}\phi(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ を求める方法を示す．$\boldsymbol{z}_p(t)=\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ とおくと，感度方程式

$$\frac{d}{dt}\boldsymbol{z}_p(t)=\frac{\partial}{\partial \boldsymbol{x}}\boldsymbol{f}(\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}),\boldsymbol{p})\boldsymbol{z}_p(t) + \frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{f}(\boldsymbol{\phi}(t,\boldsymbol{x}_{\gamma_0},\boldsymbol{p}), \quad \boldsymbol{z}_p(0)=\boldsymbol{0} \tag{26}$$

を導くことがでる [6]．$0\le t_{ev}\le T$ が成り立つので，この感度方程式 (26) を時刻 T まで解くことで $\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev},\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\phi(T,\boldsymbol{x}_{\gamma_0},\boldsymbol{p})$ の値が求められる．

一方，随伴方程式を導入して求める場合は以下の様になる [5]．新たな変数 $\hat{\boldsymbol{z}}_p$ を導入し，(26) 式の両辺と内積をとり，さらに，新たな変数 $\hat{\boldsymbol{u}}_p$ を導入して両辺に $<\boldsymbol{z}_{p_i},\hat{\boldsymbol{u}}_p>$ を加える．このとき $\frac{d}{dt}<\boldsymbol{z}_{p_i},\hat{\boldsymbol{z}}_p>=<\frac{d}{dt}\boldsymbol{z}_{p_i},\hat{\boldsymbol{z}}_p>+<\boldsymbol{z}_{p_i},\frac{d}{dt}\hat{\boldsymbol{z}}_p>$ が成り立つことと $<D\boldsymbol{f}\cdot\boldsymbol{z}_{p_i},\hat{\boldsymbol{z}}_p>=<\boldsymbol{z}_{p_i},D\boldsymbol{f}'\cdot\hat{\boldsymbol{z}}_p>$ が成り立つことに注意すると，最終的に次式が得られる．

$$<\boldsymbol{z}_{p_i},\frac{d}{dt}\hat{\boldsymbol{z}}_p+D\boldsymbol{f}'\cdot\hat{\boldsymbol{z}}_p+\hat{\boldsymbol{u}}_p> =<\boldsymbol{z}_{p_i},\hat{\boldsymbol{u}}_p>-<\frac{\partial}{\partial p_i}\boldsymbol{f},\hat{\boldsymbol{z}}_p>+\frac{d}{dt}<\boldsymbol{z}_{p_i},\hat{\boldsymbol{z}}_p> \tag{27}$$

ここで (27) 式の左辺に着目し

$$\frac{d}{dt}\hat{\boldsymbol{z}}_p=-D\boldsymbol{f}'\cdot\hat{\boldsymbol{z}}_p-\hat{\boldsymbol{u}}_p \tag{28}$$

とおき，さらに $\hat{\boldsymbol{z}}_p(T)=\boldsymbol{0}$，$\hat{\boldsymbol{u}}_p(t)=\boldsymbol{e}_j\delta(T)$ とおき，$\boldsymbol{z}_{p_i}(0)=\boldsymbol{0}$ が成り立つことに注意して両辺を $(0,T)$ の区間で積分することで次式を導くことができる．

$$\left\{\boldsymbol{z}_{p_i}(T)\right\}_j=\int_0^T <\frac{\partial}{\partial p_i}\boldsymbol{f},\hat{\boldsymbol{z}}_p>dt \tag{29}$$

(28) 式において $\tau = T - t$ とおき，入力にあたる部分 $\boldsymbol{e}_j\delta(0)$ は $\hat{\boldsymbol{z}}_p(\tau)$ に初期値を設定することと等価であることに注意すると，$\frac{\partial}{\partial \boldsymbol{p}}\phi(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めるための随伴方程式は，以下の様になる．

$$\frac{d}{d\tau}\hat{\boldsymbol{z}}_p(\tau) = D\boldsymbol{f}' \cdot \hat{\boldsymbol{z}}_p(\tau), \quad \hat{\boldsymbol{z}}_p(0) = \boldsymbol{e}_j, \quad j = 1, 2, \cdots, n \tag{30}$$

よって，随伴方程式は $\boldsymbol{z}_{p_i}(i = 1, 2, \cdots, p)$ に対しすべて同じである．さらに (24) 式の随伴方程式と同じであることから，ニュートン法が収束したときに求められた $\boldsymbol{z}_{x_0}(\tau), 0 \leq \tau \leq T$ を用いて (29) 式の積分を行えばよい．

同様にして，随伴方程式を導入して $\frac{\partial}{\partial \boldsymbol{p}}\phi(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めることができる．このときの随伴方程式は (30) 式と同じであるが，時刻 τ の定義が異なることに注意する必要がある．すなわち，$\tau = t_{ev} - t$ と定義され，$\tau = 0$，すなわち $t = t_{ev}$ のときの $\hat{\boldsymbol{z}}_p(0)$ を $\hat{\boldsymbol{z}}_p(0) = \boldsymbol{e}_j$ とおいて随伴方程式 (30) を解き，その解 $\hat{\boldsymbol{z}}_p$ を用いて次式を計算することで求められる．

$$\{\boldsymbol{z}_{p_i}(t_{ev})\}_j = \int_0^{t_{ev}} < \frac{\partial}{\partial p_i}\boldsymbol{f}, \hat{\boldsymbol{z}}_p > dt \tag{31}$$

3.3 方法の比較

感度方程式に基づく方法では，$\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めるために，(23) 式の n 次元の感度方程式を $t = 0$ から $t = T$ まで n 回解き，$\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めるために，(26) 式の n 次元の感度方程式を $t = 0$ から $t = T$ まで p 回解く．

一方，随伴方程式に基づく方法では，$\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(T, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めるために，$\tau = T - t$ とおいて (24) 式の n 次元の随伴方程式を n 回解き，(29) 式の積分を $n \times p$ 回行う．さらに，$\frac{\partial}{\partial \boldsymbol{x}_0}\boldsymbol{\phi}(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ と $\frac{\partial}{\partial \boldsymbol{p}}\boldsymbol{\phi}(t_{ev}, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を求めるために，$\tau = t_{ev} - t$ とおいて (24) 式の n 次元の随伴方程式を n 回解き，(31) 式の積分を $n \times p$ 回行う．

以上のことから，積分を行うことと微分方程式を解くことは本質的に同じであることを考慮すると，随伴方程式に基づく方法の計算量は感度方程式に基づく方法のそれの 2 倍程度である．

感度方程式を解く際，対象のシステムの解 $\boldsymbol{\phi}(t, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ が必要となるが，システムの方程式とこの感度方程式は連立して同時に解くことができる．一方，随伴方程式を解く際も対象のシステムの解 $\boldsymbol{\phi}(t, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ が必要となるが，随伴方程式は時刻 T から 0 へ，あるいは時刻 t_{ev} から 0 へ解くことから，システムの解 $\boldsymbol{\phi}(t, \boldsymbol{x}_{\gamma_0}, \boldsymbol{p})$ を保存しておく必要がある．

4 感度の計算例

本節では，感度方程式に基づく方法と随伴方程式に基づく方法とでそれぞれ位相感度と周期感度を求めた計算例を示す．最初の例は，その感度を解析的に求めることができる 2 次のシステムを対象とし，提案法の精度を厳密に評価する．次の例は，Rössler System と呼ばれるそのパラメータの値によってカオスを発生するシステムを対象とする．このシステムの安定なリミットサイクル，およびカオスアトラクターに埋め込まれた不安定なリミットサイクルを対象として提案法を用いて感度解析を行う．最後に，生体システムの代表的なリズム現象の一つである概日リズムのモデルに対し提案法を適用する．なお，最初の例以外はその感度が解析的には求められないので，差分近似によって求めた感度と提案法の結果と比較することによりその性能を評価する．

4.1 位相感度が解析的に求められる例

次の二次のシステムを対象とする．

$$\begin{cases} \frac{d}{dt}x_1(t) = p_2 x_1(t) - p_1 x_2(t) - (x_1^2(t) + x_2^2(t))x_1(t) \\ \frac{d}{dt}x_2(t) = p_1 x_1(t) + p_2 x_2(t) - (x_1^2(t) + x_2^2(t))x_2(t) \end{cases} \tag{32}$$

ここでのパラメータは $\boldsymbol{p} = [p_1 \ p_2]'$ であり，これらを $\boldsymbol{p} = [1.0 \ 1.0]'$ と与えることにする．このシステムは解析解を求めることができ，$p_1 \neq 0, p_2 > 0$ のとすると，$x_0 = [\sqrt{p_2} \ 0]'$ を初期状態とする解は

$$\begin{cases} \phi_1(t, \boldsymbol{x}_0, \boldsymbol{p}) & = \sqrt{p_2}\cos(p_1 t) \\ \phi_2(t, \boldsymbol{x}_0, \boldsymbol{p}) & = \sqrt{p_2}\sin(p_1 t) \end{cases} \tag{33}$$

と表され，周期 $\frac{2\pi}{p_1}$ の周期解となる．ここでは，周期軌道において初期状態 $x_0 = [\sqrt{p_2} \ 0]'$ を位相を測る基準として，すなわちこの点を位相 0 として位相感度を求める．位相を求める事象として，$x_1(t) = -\sqrt{\frac{p_2}{2}}$ となる事象，および $\frac{d}{dt}x_2(t) = 0$ となる事象とし，すなわち $\mathcal{E}_{v1}(\boldsymbol{x}, \boldsymbol{p}) = x_1(t) + \sqrt{\frac{p_2}{2}}$，および $\mathcal{E}_{v2}(\boldsymbol{x}, \boldsymbol{p}) = p_1 x_1(t) + p_2 x_2(t) - (x_1^2(t) + x_2^2(t))x_2(t)$ とし，それぞれの事象の起こる時刻を t_{ev1}，t_{ev2}，またそれらの位相を θ_1，θ_2 とする．ここでは，このときの位相感度 $\frac{\partial \theta_1}{\partial \boldsymbol{p}}$，$\frac{\partial \theta_2}{\partial \boldsymbol{p}}$ を求める．この場合，事象の起こる時刻，およびその位相は解析的に求めることができ，それぞれ，$t_{ev1} = \frac{3\pi}{4p_1}$，$t_{ev2} = \frac{\pi}{2p_1}$，$\theta_1 = \frac{3\pi}{4}$，$\theta_2 = \frac{\pi}{2}$ と求まる．これらより，周期感度は $\frac{\partial T}{\partial \boldsymbol{p}} = \begin{bmatrix} -\frac{2\pi}{p_1^2} & 0 \end{bmatrix}'$，$t_{ev1}$ の $\boldsymbol{p}$ に関する感度は $\frac{\partial t_{ev1}}{\partial \boldsymbol{p}} = \begin{bmatrix} -\frac{3\pi}{4p_1^2} & 0 \end{bmatrix}'$，$t_{ev2}$ の $\boldsymbol{p}$ に関する感度は $\frac{\partial t_{ev2}}{\partial \boldsymbol{p}} = \begin{bmatrix} -\frac{\pi}{2p_1^2} & 0 \end{bmatrix}'$，位相感度はそれぞれ $\frac{\partial \theta_1}{\partial \boldsymbol{p}} = \begin{bmatrix} 0 & 0 \end{bmatrix}'$，$\frac{\partial \theta_2}{\partial \boldsymbol{p}} = \begin{bmatrix} 0 & 0 \end{bmatrix}'$ と求まる．

以上のように解析解が求まる例に対し，提案法を次のように適用し感度を求めた．まず，ポアンカレ断面 Σ を $\sigma(\boldsymbol{x}) = x_2 = 0$ として 3.2.1 項で説明した方法により周期軌道のポアンカレ断面 Σ との交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T を求めると，$\boldsymbol{x}_{\gamma_0} = [1.00000 \ 0.00000]'$，$T = 6.28319$ と得られ，また，$t_{ev1} = 2.35619$,$t_{ev2} = 1.57080$ と得られた．Table 1 に，これらに対し提案法を適用し感度を求めた結果と解析的に得られた真値を示す．この Table より，提案法で求めた感度の値はそれらの真値と一致しており，提案法で正しく感度が求めらることがわかる．

Table 1: Values of sensitivities for the periodic orbit of (33) obtained by using the proposed method and their true values

Sensi-tivities	Obtained values		True values
	Sensitivity equations	Adjoint equations	
$\frac{\partial T}{\partial p_1}$	-6.28319	-6.28319	-2π
$\frac{\partial T}{\partial p_2}$	5.37288×10^{-8}	-1.98213×10^{-11}	0
$\frac{\partial t_{ev1}}{\partial p_1}$	-2.35619	-2.35619	$-\frac{3\pi}{4}$
$\frac{\partial t_{ev1}}{\partial p_2}$	2.50500×10^{-9}	-1.54378×10^{-9}	0
$\frac{\partial t_{ev2}}{\partial p_1}$	-1.57080	-1.57080	$-\frac{\pi}{2}$
$\frac{\partial t_{ev2}}{\partial p_2}$	2.03079×10^{-9}	-2.78651×10^{-9}	0
$\frac{\partial \theta_1}{\partial p_1}$	-1.31744×10^{-8}	3.45821×10^{-8}	0
$\frac{\partial \theta_1}{\partial p_2}$	-5.26239×10^{-9}	-1.54342×10^{-9}	0
$\frac{\partial \theta_1}{\partial p_1}$	-1.30390×10^{-7}	2.38635×10^{-7}	0
$\frac{\partial \theta_1}{\partial p_2}$	-1.14014×10^{-8}	-2.78627×10^{-8}	0

4.2　安定なリミットサイクルの例

Rössler System と呼ばれる次式で表されるシステム [10] を対象とする．

$$\begin{cases} \frac{d}{dt}x_1(t) = -x_2(t) - x_3(t) \\ \frac{d}{dt}x_2(t) = x_1(t) + p_1 x_2(t) \\ \frac{d}{dt}x_3(t) = p_2 + x_3(t)(x_1(t) - p_3) \end{cases} \tag{34}$$

このシステムは，パラメータ $\boldsymbol{p} = [p_1\ p_2\ p_3]'$ の値により，カオスを発生することが知られている [10]．ここではまずこのシステムの安定なリミットサイクルを対象とする．パラメータ $\boldsymbol{p}$ を $\boldsymbol{p} = [0.2\ 0.2\ 2.6]'$ とし，ポアンカレ断面 Σ を $\sigma(\boldsymbol{x}) = x_3 - 1.0 = 0$ とし，3.2.1 項で説明した方法により，周期軌道のポアンカレ断面 Σ との交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T を求めると，それぞれ $\boldsymbol{x}_{\gamma_0} = [4.893635\ \ -0.8806169\ 1.000000]'$, $T = 5.755597$ と得られた．Fig. 1 に求められた周期解軌道を示す．また，このときに得られたヤコビ行列 $D\boldsymbol{\phi}(T, \boldsymbol{x}_\gamma, \boldsymbol{p})$ は，

$$D\boldsymbol{\phi} = \begin{bmatrix} 0.5442572 & 0.8590171 & -1.646925 \\ 0.9554855 & -0.05122281 & 2.034469 \\ 1.007236 & 0.7306589 & -0.3340546 \end{bmatrix}$$

で，その固有値は，$[1.000000,\ \ -2.663039 \times 10^{-6},\ -0.8410176]$ であった．したがって，求められた周期解は安定なリミットサイクルであると判別できる．このリミットサイクルを対象として，軌道が $x_2 = 2.0$ となるという事象が生じるとき，すなわち $\mathcal{E}_{v1}(\boldsymbol{x}, \boldsymbol{p}) = x_2 - 2.0$ としたとき，および $\frac{d}{dt}x_2(t) = 0$ となるという事象が生じるとき，すなわち $\mathcal{E}_{v2}(\boldsymbol{x}, \boldsymbol{p}) = x_1(t) + p_1 x_2(t)$ としたときの位相のパラメータ感度 $\frac{\partial \theta_1}{\partial \boldsymbol{p}}$ および $\frac{\partial \theta_2}{\partial \boldsymbol{p}}$ を提案法により求めることにする．なお，t_{ev1}, θ_1, t_{ev2}, θ_2 はそれぞれ，$t_{ev1} = 0.8496555$, $\theta_1 = 0.9275393$, $t_{ev2} = 1.343512$, $\theta_2 = 1.466666$ と得られた．またこれらの感度を解析的に求めるのは困難であるので，提案法の結果の妥当性を検証するために，差分近似により求めた感度と比較することにする．$p_i, i = 1,2,3$ をそれぞれ $\Delta p_i = 0.001$ だけ変化させたときの周期 $\tilde{T}$ とポアンカレ断面との交点 $\tilde{\boldsymbol{x}}_{\gamma_0}$ を 3.2.1 項で説明した方法で求め，変化後の周期軌道 $\tilde{\boldsymbol{\gamma}}$ を求める．これより $\tilde{\boldsymbol{\gamma}}$ の位相 $\tilde{\theta}_j$, $j = 1,2$ を求め，それらの値を基にして，次式で位相感度の近似値 $\frac{\Delta \theta_j}{\Delta p_i}$, $j = 1,2$ を求める．

$$\frac{\Delta \theta_j}{\Delta p_i} = \frac{\tilde{\theta}_j - \theta_j}{\Delta p_i}, \quad i = 1,2,3, j = 1,2 \tag{35}$$

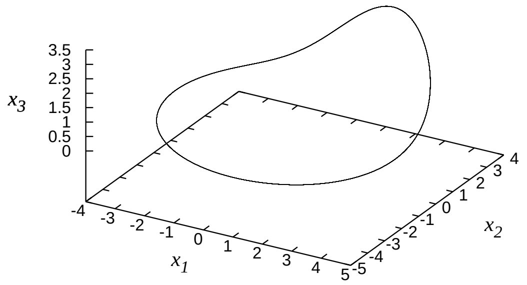


Fig. 1: Obtained stable limit cycle of Rössler system (34).

Table 2: Values of sensitivities of stable limit cycle obtained by the using proposed method and the difference method

Sensi-tivities	Obtained values		
	Sensitivity equations	Adjoint equations	Deffernce method
$\frac{\partial T}{\partial p_1}$	-2.63884	-2.63884	-2.63678
$\frac{\partial T}{\partial p_2}$	0.991910	0.991910	0.989833
$\frac{\partial T}{\partial p_3}$	0.0644745	0.0644745	0.0644700
$\frac{\partial t_{ev1}}{\partial p_1}$	0.928815	0.928815	0.920609
$\frac{\partial t_{ev1}}{\partial p_2}$	0.162657	0.162657	0.161540
$\frac{\partial t_{ev1}}{\partial p_3}$	0.154795	0.154795	0.154752
$\frac{\partial t_{ev2}}{\partial p_1}$	0.325562	0.325562	0.318881
$\frac{\partial t_{ev2}}{\partial p_2}$	1.06763	1.06763	1.06561
$\frac{\partial t_{ev2}}{\partial p_3}$	0.0695680	0.0695680	0.069558
$\frac{\partial \theta_1}{\partial p_1}$	1.43921	1.43921	1.43058
$\frac{\partial \theta_1}{\partial p_2}$	0.0177167	0.0177167	0.0168287
$\frac{\partial \theta_1}{\partial p_3}$	0.158594	0.158594	0.158547
$\frac{\partial \theta_2}{\partial p_1}$	1.02784	1.02784	1.02050
$\frac{\partial \theta_2}{\partial p_2}$	0.912734	0.912734	0.910900
$\frac{\partial \theta_2}{\partial p_3}$	0.0595153	0.0595153	0.059505

事象の起こる時刻に対する感度など他の感度の近似値も同様に求める．Table 2 に，以上により求められた結果を示す．この結果より，提案法により適切に感度が求められていることが分かる．

4.3　不安定なリミットサイクルの例

先にも述べたように，(34) 式で表される Rössler System はカオスを発生することがある．たとえばパラメータ $\boldsymbol{p}$ を $\boldsymbol{p} = [0.2\ 0.2\ 5.8]'$ とすると，システムは Fig. 2 に示すようなカオスアトラクターをもつ．ここでは，このカオスアトラクターに埋め込まれている不安定なリミットサイクルを対象とする．そこで，3.2.1 項で説明した方法で不安定なリミットサイクルを求める．ポアンカレ断面を $\sigma(\boldsymbol{x}) = x_1 - 7.0 = 0$ とすると，リミットサイクルとポアンカレ断面との交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T の値がそれぞれ $\boldsymbol{x}_{\gamma_0} = [7.000000\ 3.245391\ 7.073312]'$, $T = 5.883814857$ のように得られた．得られたリミットサイクルを Fig. 3 に示す．また，このときに得られたヤコビ行列 $D\boldsymbol{\phi}(T, \boldsymbol{x}_\gamma, \boldsymbol{p})$ は，

$$D\boldsymbol{\phi} = \begin{bmatrix} -0.4955839 & -1.771688 & -0.2164725 \\ -0.04617354 & 0.5178315 & 0.3696711 \\ 3.666980 & 7.742870 & -1.461715 \end{bmatrix}$$

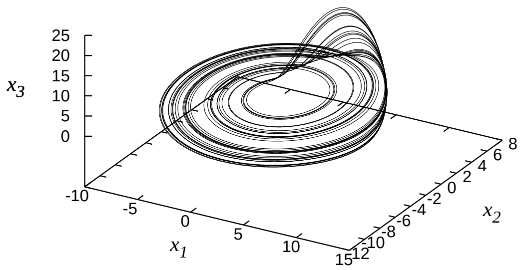


Fig. 2: Chaos attracter of Rössler system (34).

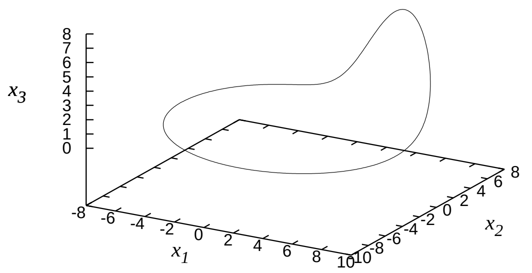


Fig. 3: Obtained unstable limit cycle embedded in the chaos atracter of Fig. 2.

で，その固有値は，-2.439468, 3.647235×10^{-12}, 1.000000 であり，得られたリミットサイクルは不安定であることがわかる．このリミットサイクルを対象として，軌道が $x_3(t) - 7.0 = 0$ となるという事象が生じるとき，すなわち $\mathcal{E}_{v1}(\boldsymbol{x}, \boldsymbol{p}) = x_3(t) - 7.0$ としたとき，および $\frac{d}{dt}x_3(t) = 0$ となるという事象が生じるとき，すなわち $\mathcal{E}_{v2}(\boldsymbol{x}, \boldsymbol{p}) = p_2 + x_3(t)(x_1(t) - p_3)$ としたときの位相 θ_1, θ_2 のパラメータ感度 $\frac{\partial \theta_1}{\partial \boldsymbol{p}}$ および $\frac{\partial \theta_2}{\partial \boldsymbol{p}}$ を求めることにする．なお，t_{ev1}, t_{ev2}，θ_1，θ_2 はぞれぞれ，$t_{ev1} = 0.2274579$, $t_{ev2} = 0.1108945$，$\theta_1 = 0.2428969$, $\theta_2 = 0.1184216$ と得られた．Table 3 に，提案法によって求められた感度の値と，差分近似により得られた感度の値を示す．この Table より，提案法により適切に感度が求められていることが分かる．

4.4　ショウジョウバエの概日リズムの例

ここでは，生体システムの概日リズムの研究でよくとりあげられている次式で表されるショウジョウバエの概日リズムを対象とする [11]．

$$
\begin{cases}
\frac{d}{dt}x_1(t) &= \frac{p_1}{1+\left(\frac{x_2(t)(1-q)}{2p_8}\right)^2} - p_2 x_1(t) \\
\frac{d}{dt}x_2(t) &= p_3 x_1(t) - \frac{(p_4 q + p_5) x_2(t)}{p_9 + x_2(t)} - p_6 x_2(t) \\
q = & \frac{2}{1+\sqrt{1+8p_7 x_2(t)}}
\end{cases}
\quad (36)
$$

このシステムは，パラメータを $\boldsymbol{p}$=$[p_1\ p_2\ p_3\ p_4\ p_5\ p_6\ p_7\ p_8\ p_9]' = [1\ 0.1\ 0.5\ 10\ 0.03\ 0.1\ 200\ 0.1\ 0.05]'$ としたとき，約 24 時間の周期軌道をもつ．ポアンカレ断面を $\sigma(\boldsymbol{x}) = x_2(t) - 2.0 = 0$ とし，3.2.1 項で説明した

Table 3: Values of sensitivities of unstable limit cycle obtained by using the proposed method and difference method

Sensi-tivities	Obtained values		
	Sensitivity equations	Adjoint equations	Deffernce method
$\frac{\partial T}{\partial p_1}$	-1.55353	-1.55353	-1.55069
$\frac{\partial T}{\partial p_2}$	0.527629	0.527628	0.526560
$\frac{\partial T}{\partial p_3}$	0.0270276	0.270277	0.0271797
$\frac{\partial t_{ev1}}{\partial p_1}$	3.40010	3.40010	3.32181
$\frac{\partial t_{ev1}}{\partial p_2}$	-0.817278	-0.817277	-0.821047
$\frac{\partial t_{ev1}}{\partial p_3}$	0.0110913	0.0110912	0.0110201
$\frac{\partial t_{ev2}}{\partial p_1}$	-0.351816	-0.351816	-0.350368
$\frac{\partial t_{ev2}}{\partial p_2}$	0.187190	0.187190	0.187067
$\frac{\partial t_{ev2}}{\partial p_3}$	-0.115731	-0.115731	-0.115695
$\frac{\partial \theta_1}{\partial p_1}$	3.69502	3.69502	3.61225
$\frac{\partial \theta_1}{\partial p_2}$	-0.894533	-0.894532	-0.898434
$\frac{\partial \theta_1}{\partial p_3}$	0.0107284	0.0107282	0.0106524
$\frac{\partial \theta_2}{\partial p_1}$	-0.344429	-0.344429	-0.343030
$\frac{\partial \theta_2}{\partial p_2}$	0.189276	0.189276	0.189150
$\frac{\partial \theta_2}{\partial p_3}$	-0.124130	-0.124130	-0.124091

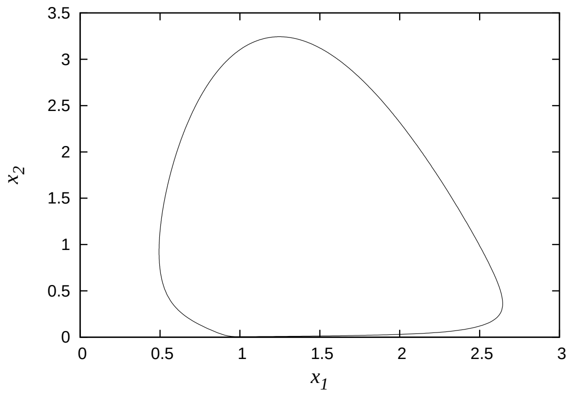


Fig. 4: A periodic trajectory of *Drosophila* circadian rhythm (36).

方法により，リミットサイクルと周期軌道のポアンカレ断面 Σ との交点 $\boldsymbol{x}_{\gamma_0}$ と周期 T を求めると，それぞれ $\boldsymbol{x}_{\gamma_0} = [2.133354\ 2.000000]'$, $T = 24.20420$ と得られた．Fig. 4 に得られたリミットサイクルを示す．また，このときに得られたヤコビ行列 $D\boldsymbol{\phi}(T, \boldsymbol{x}_\gamma, \boldsymbol{p})$ は，

$$
D\boldsymbol{\phi} = \begin{bmatrix} 1.570873 & 0.2316894 \\ -3.870568 & -0.5708732 \end{bmatrix}
$$

で，その固有値は，1.000000, $-3.019807 \times 10^{-13}$ であった．したがって，求められた周期解は安定なリミットサイクルであると判別できる．このリミットサイクルを対象として，軌道が $x_1(t) - 1.0 = 0$ となる事象が生じるとき，すなわち $\mathcal{E}_{v1}(\boldsymbol{x}, \boldsymbol{p}) = x_1(t) - 1.0$ としたとき，および $\frac{d}{dt}x_1(t) = 0$ となるとなる事象が生じるとき，すなわち $\mathcal{E}_{v2}(\boldsymbol{x}, \boldsymbol{p}) = \frac{p_1}{1+\left(\frac{x_2(t)(1-q)}{2p_8}\right)^2} - p_2 x_1(t)$ としたときの位相 θ_1, θ_2 のパラメータ感度 $\frac{\partial \theta_1}{\partial p}$ および $\frac{\partial \theta_2}{\partial p}$ を求めることにする．なお，t_{ev1}, t_{ev2}，θ_1，θ_2 はそれぞれ，$t_{ev1} = 7.861385$, $t_{ev2} = 16.90535$，$\theta_1 = 2.040743$, $\theta_2 = 4.388472$ と得られた．Table 4 に，提案法によって求められた感度の値と，差分近似により得られた感度の値を示す．この Table より，提案法により適切に感度が求められていることが分かる．

Table 4: Values of sensitivities of *Drosophila* circadian rhythm obtained by the proposed method and the difference method

Sensitivities	Obtained values		
	Sensitivity equations	Adjoint equations	Deffernce method
$\frac{\partial T}{\partial p_1}$	2.53174	2.53174	2.53240
$\frac{\partial T}{\partial p_2}$	−188.158	−188.158	−186.704
$\frac{\partial T}{\partial p_3}$	5.06350	5.06349	5.06557
$\frac{\partial T}{\partial p_4}$	−0.0403334	−0.0403327	−0.0402900
$\frac{\partial T}{\partial p_5}$	−31.8281	−31.8281	−31.7915
$\frac{\partial T}{\partial p_6}$	−90.9368	−90.9367	−90.3936
$\frac{\partial T}{\partial p_7}$	0.00284568	0.00284567	0.00286000
$\frac{\partial T}{\partial p_8}$	48.1462	48.1461	48.0188
$\frac{\partial T}{\partial p_9}$	−108.381	−108.381	−107.108
$\frac{\partial t_{ev1}}{\partial p_1}$	3.73801	3.73801	3.73589
$\frac{\partial t_{ev1}}{\partial p_2}$	−123.513	−123.513	−122.518
$\frac{\partial t_{ev1}}{\partial p_3}$	−13.4394	−13.4395	−13.4262
$\frac{\partial t_{ev1}}{\partial p_4}$	0.770412	0.770412	0.770697
$\frac{\partial t_{ev1}}{\partial p_5}$	2.15043	2.15044	2.14849
$\frac{\partial t_{ev1}}{\partial p_6}$	−2.96962	−2.96960	−2.97969
$\frac{\partial t_{ev1}}{\partial p_7}$	−0.0177469	−0.0177469	−0.0178720
$\frac{\partial t_{ev1}}{\partial p_8}$	33.7797	33.7797	33.6037
$\frac{\partial t_{ev1}}{\partial p_9}$	−79.5924	−79.5924	−79.0415
$\frac{\partial t_{ev2}}{\partial p_1}$	5.28379	5.28379	5.28043
$\frac{\partial t_{ev2}}{\partial p_2}$	−193.666	−193.666	−192.400
$\frac{\partial t_{ev2}}{\partial p_3}$	10.5676	10.5676	10.5530
$\frac{\partial t_{ev2}}{\partial p_4}$	0.162262	0.162263	0.162340
$\frac{\partial t_{ev2}}{\partial p_5}$	−30.1974	−30.1973	−30.1747
$\frac{\partial t_{ev2}}{\partial p_6}$	−88.2306	−88.2306	−87.7720
$\frac{\partial t_{ev2}}{\partial p_7}$	−0.00179548	−0.00179549	−0.00181000
$\frac{\partial t_{ev2}}{\partial p_8}$	39.1695	39.1695	38.8628
$\frac{\partial t_{ev2}}{\partial p_9}$	−125.598	−125.598	−124.634
$\frac{\partial \theta_1}{\partial p_1}$	0.756892	0.756892	0.756208
$\frac{\partial \theta_1}{\partial p_2}$	−16.1986	−16.1986	−16.1877
$\frac{\partial \theta_1}{\partial p_3}$	−3.91568	−3.91568	−3.91160
$\frac{\partial \theta_1}{\partial p_4}$	0.203392	0.203392	0.203463
$\frac{\partial \theta_1}{\partial p_5}$	3.24178	3.24178	3.24244
$\frac{\partial \theta_1}{\partial p_6}$	6.89632	6.89632	6.87358
$\frac{\partial \theta_1}{\partial p_7}$	−0.00484685	−0.00484685	−0.00488100
$\frac{\partial \theta_1}{\partial p_8}$	4.70951	4.70951	4.66531
$\frac{\partial \theta_1}{\partial p_9}$	−11.5235	−11.5235	−11.5388
$\frac{\partial \theta_2}{\partial p_1}$	0.912592	0.912593	0.911506
$\frac{\partial \theta_2}{\partial p_2}$	−16.1588	−16.1588	−16.2190
$\frac{\partial \theta_2}{\partial p_3}$	1.82519	1.82519	1.82065
$\frac{\partial \theta_2}{\partial p_4}$	0.0494347	0.0494347	0.0494460
$\frac{\partial \theta_2}{\partial p_5}$	−2.06818	−2.06818	−2.07168
$\frac{\partial \theta_2}{\partial p_6}$	−6.41607	−6.41607	−6.41947
$\frac{\partial \theta_2}{\partial p_7}$	−0.000982042	−0.000982044	−0.000989000
$\frac{\partial \theta_2}{\partial p_8}$	1.43864	1.43864	1.37938
$\frac{\partial \theta_2}{\partial p_9}$	−12.9536	−12.9536	−12.9914

5 おわりに

リズム現象は，ありとあらゆるシステムにみられる興味深い非線形現象で，その発生メカニズムを解明し，その成果を利用してシステムの解析，設計における種々の問題の解決を図ろうとする研究が盛んになっている．本稿では，非線形自律システムに現れるリズム現象を対象としてその感度解析法について議論し，特に，リズムの位相および周期のパラメータに関する感度を求める方法を提案した．そのために，システムの周期軌道の位相の定義を与え，位相と周期に対するパラメータ感度の厳密な表現を導いた．また導いた表現に基づき感度を計算するための精度が高くかつ計算効率がよい計算アルゴリズムを導出した．提案した方法は，安定なリミットサイクルの周期感度だけなく，たとえばカオスアトラクターに埋め込まれた不安定なリミットサイクルの周期感度も求めることできる方法となっている．

参考文献

1) D. Edelson, V. M. Thomas: Sensitivity Analysis of Oscillating Reactions. 1. The Period of the Oregonator, J. Phys. Chem., **85**-11, 1555/1558 (1981)

2) M. A. Kramer, H. Rabitz, J. M. Calo: Sensitivity Analysis of Oscillatory Systems. Appl Math Modelling, **8**, 328/340 (1984)

3) A. K. Wilkins, B. Tidor, J. White, P. I. Barton: Sensitivity Analysis for Oscillating Dynamical Systems, SIAM Journal on Scientific Computing, **31**-4, 2706/2732 (2009)

4) 黒江，森：リズム現象における周期感度の解析法，計測自動制御学会第 5 回コンピューテーショナル・インテリジェンス研究会資料，31/36 (2014)

5) 森，黒江：リズム現象における周期のパラメータ感度の解析法，第 57 回自動制御連合講演会講演論文集，1145/1152, (2014)

6) 黒江，森：リズム現象に対するパラメータ感度の解析法，第 58 回自動制御連合講演会講演論文集，2D3-1, 8 pages, (2015)

7) J. Guckenheimer and P. Holms, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Field.* New York: Springer-Verlag (1983)

8) 黒江：ダイナミカルシステム入門，計測自動制御学会誌 計測と制御，**46**-4, 230/239 (2007)

9) Y. Kuroe: Computer-Aided Design Method of Stabilizing Controllers for Chaotic Systems, Proceedings of 2012 IEEE International Symposium on Intelligent Control (ISIC) Part of 2012 IEEE Multi-Conference on Systems and Control, 282/288 (2012)

10) A. T. Alligood, T. D. Sauer and J.A. York: *Chaos, An Introduction to Dynamical Systems*, New York, Springer-Verlag (1997)

11) J. J. Tyson, C. I. Hong, C. D. Thron and B. Novak: A Simple Model of Circadian Rhythms Based on Dimerization and Proteolysis of PER and TIM, Biophysical Jounal, **77**, 2411/2417 (1999)

Riemannian preconditioning for tensor completion

○笠井 裕之 (電気通信大学)* Bamdev Mishra (Amazon Development Centre India) †

Riemannian preconditioning for tensor completion

Hiroyuki Kasai (The University of Electro-Communications)
Bamdev Mishra (Amazon Development Centre India)

Abstract– We propose a novel Riemannian preconditioning approach for the tensor completion problem with rank constraint. A Riemannian metric or inner product is proposed that exploits the least-squares structure of the cost function and takes into account the structured symmetry in Tucker decomposition. The specific metric allows to use the versatile framework of Riemannian optimization on quotient manifolds to develop a preconditioned nonlinear con ugate gradient algorithm for the problem. Numerical comparisons suggest that our proposed algorithm robustly outperforms state-of-the-art algorithms across different problem instances encompassing various synthetic and real-world datasets.

Key Words: Riemannian optimization, Structured symmetry, Least-squares, Tailored Riemannian metric, Tucker decomposition

1 Introduction

This paper addresses the problem of low-rank tensor completion when the rank is a priori known or estimated. Without loss of generality, we focus on 3-order tensors. Given a tensor $\boldsymbol{\mathcal{X}}^{n_1\times n_2\times n_3}$, whose entries $\boldsymbol{\mathcal{X}}^\star_{i_1,i_2,i_3}$ are only known for some indices $(i_1, i_2, i_3) \in \Omega$, where Ω is a subset of the complete set of indices $\{(i_1, i_2, i_3) : i_d \in \{1, \ldots, n_d\}, d \in \{1, 2, 3\}\}$, the *fixed-rank tensor completion problem* is formulated as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\mathcal{X}}\in\mathbb{R}^{n_1\times n_2\times n_3}} & \frac{1}{|\Omega|}\|\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star)\|_F^2 \\ \text{subject to} & \operatorname{rank}(\boldsymbol{\mathcal{X}}) = \mathbf{r}, \end{array} \tag{1}$$

where the operator $\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}})_{i_1i_2i_3} = \boldsymbol{\mathcal{X}}_{i_1i_2i_3}$ if $(i_1, i_2, i_3) \in \Omega$ and $\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}})_{i_1i_2i_3} = 0$ otherwise and (with a slight abuse of notation) $\|\cdot\|_F$ is the Frobenius norm. $\operatorname{rank}(\boldsymbol{\mathcal{X}})$ $(= \mathbf{r} = (r_1, r_2, r_3))$, called the *multilinear rank* of $\boldsymbol{\mathcal{X}}$, is the set of the ranks of for each of mode-d unfolding matrices. $r_d \ll n_d$ enforces a low-rank structure. The *mode* is a matrix obtained by concatenating the mode-d fibers along column and mode-d *unfolding* of $\boldsymbol{\mathcal{X}}$ is $\mathbf{X}_d \in \mathbb{R}^{n_d\times n_{d+1}\cdots n_D n_1\cdots n_{d-1}}$ for $d = \{1, \ldots, D\}$.

The optimization problem (1) has many variants, and one of those is extending the nuclear norm regularization approach from the matrix case [1] to the tensor case. While this generalization leads to good results [2, 3, 4], its scalabilityto large-scale instances is not trivial, especially due to the necessity of high-dimensional singular value decomposition computations. A different approach exploits *Tucker decomposition* [5, Section 4] of a low-rank tensor $\boldsymbol{\mathcal{X}}$ to develop large-scale algorithms for (1), e.g., in [6, 7]. The present paper exploits both the *symmetry* present in Tucker decomposition and the *least-squares* structure of the cost function of (1) by using the concept of *preconditioning*. While preconditioning in unconstrained optimization is well studied [8, Chapter 5], preconditioning on constraints with *symmetries*, owing to non-uniqueness of Tucker decomposition [5, Section 4.3], is not straightforward. We build upon the recent work [9] that suggests to use *Riemannian preconditioning* with a *tailored metric* (inner product) in the Riemannian optimization framework on quotient manifolds [10, 11, 12]. Our proposed preconditioned nonlinear conjugate gradient algorithm is implemented in the Matlab toolbox Manopt [13] and it outperforms state-of-the-art methods. In the supplementary material section, we show concrete mathematical derivations and additional numerical comparisons. We also provide a *generic* Manopt factory (a manifold description Matlab file) with additional support for second-order implementations, e.g., the trust-region method.

2 Exploiting the problem structure

We focus on the two fundamental structures present in (1): *symmetry* in the constraints, and the *least-squares structure* of the cost function. Finally, a novel metric is proposed.

The quotient and least-squares structures. The Tucker decomposition of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ of rank $\mathbf{r}$ $(=(r_1, r_2, r_3))$ is [5, Section 4.1] $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3$, where $\mathbf{U}_d \in \mathrm{St}(r_d, n_d)$ for $d \in \{1, 2, 3\}$ belongs to the *Stiefel manifold* of matrices of size $n_d \times r_d$ with orthogonal columns and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1\times r_2\times r_3}$. Here, $\boldsymbol{\mathcal{W}}\times_d\mathbf{V} \in \mathbb{R}^{n_1\times\cdots n_{d-1}\times m\times n_{d+1}\times\cdots n_N}$ computes the *d-mode product* of a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n_1\times\cdots\times n_N}$ and a matrix $\mathbf{V} \in \mathbb{R}^{m\times n_d}$. Tucker decomposition is *not unique* as $\boldsymbol{\mathcal{X}}$ remains un-

*This work was initiated while Hiroyuki Kasai was with the Department of Electrical and Computer Engineering, Technische Universität München, Germany.

†This work was initiated while Bamdev Mishra was with the Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium and was visiting the Department of Engineering (Control Group), University of Cambridge, Cambridge, UK.

changed under the transformation $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}}) \mapsto (\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}}\times_1\mathbf{O}_1^T\times_2\mathbf{O}_2^T\times_3\mathbf{O}_3^T)$ for all $\mathbf{O}_d \in \mathcal{O}(r_d)$, which is the set of orthogonal matrices of size of $r_d \times r_d$. The classical remedy to remove this indeterminacy is to have additional structures on $\boldsymbol{\mathcal{G}}$ like sparsity or restricted orthogonal rotations [5, Section 4.3]. In contrast, we encode the transformation in an abstract search space of *equivalence classes*, defined as, $[(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})] := \{(\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}}\times_1\mathbf{O}_1^T\times_2\mathbf{O}_2^T\times_3\mathbf{O}_3^T) : \mathbf{O}_d \in \mathcal{O}(r_d)\}$. The set of equivalence classes is the quotient manifold [14, Theorem 9.16)]

$$\mathcal{M}/\sim \quad := \quad \mathcal{M}/(\mathcal{O}(r_1)\times\mathcal{O}(r_2)\times\mathcal{O}(r_3)), \qquad (2)$$

where $\mathcal{M}$ is called the *total space* (computational space) that is the product space $\mathcal{M} := \mathrm{St}(r_1, n_1) \times \mathrm{St}(r_2, n_2) \times \mathrm{St}(r_3, n_3) \times \mathbb{R}^{r_1\times r_2\times r_3}$. Due to the invariance of the Tucker decomposition, the local minima of (1) in $\mathcal{M}$ are not isolated, but they become isolated on $\mathcal{M}/\sim$. Consequently, the problem (1) is an optimization problem on a quotient manifold for which systematic procedures are proposed in [10, 11, 12)] by endowing $\mathcal{M}/\sim$ with a Riemannian structure. We call $\mathcal{M}/\sim$ the *Tucker manifold*.

Another structure that is present in (1) is the least-squares structure of the cost function. A way to exploit it is to endow the search space with a metric (inner product) induced by the Hessian of the cost function [8)]. This induced metric (or its approximation) resolves convergence issues of first-order optimization algorithms. Specifically for the case of quadratic optimization with rank constraint (matrix case), Mishra and Sepulchre [9, Section 5)] propose a family of Riemannian metrics from the Hessian of the cost function. Since applying this approach directly for (1) is computationally costly, we consider a simplified cost function by assuming that Ω contains the full set of indices, i.e., we focus on $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ to propose a metric candidate. A good candidate is by considering only the *block diagonal* elements of the Hessian of $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$. It should emphasized that the cost function $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ is *convex and quadratic* in $\boldsymbol{\mathcal{X}}$. Consequently, it is also convex and quadratic in the arguments $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ individually. The block diagonal approximation of the Hessian of $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ in $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ is $((\mathbf{G}_1\mathbf{G}_1^T)\otimes\mathbf{I}_{n_1}, (\mathbf{G}_2\mathbf{G}_2^T)\otimes\mathbf{I}_{n_2}, (\mathbf{G}_3\mathbf{G}_3^T)\otimes\mathbf{I}_{n_3}, \mathbf{I}_{r_1r_2r_3})$, where $\mathbf{G}_d$ is the mode-d unfolding of $\boldsymbol{\mathcal{G}}$ and is assumed to be full rank. The terms $\mathbf{G}_d\mathbf{G}_d^T$ for $d \in \{1, 2, 3\}$ are *positive definite* when $r_1 \le r_2r_3$, $r_2 \le r_1r_3$, and $r_3 \le r_1r_2$.

A novel Riemannian metric and its motivation. An element x in the total space $\mathcal{M}$ has the matrix representation $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$. Consequently, the tangent space $T_x\mathcal{M}$ is the Cartesian product of the tangent spaces of the individual manifolds, i.e., $T_x\mathcal{M}$ has the matrix characterization [12)] $T_x\mathcal{M} = \{(\mathbf{Z}_{\mathbf{U}_1}, \mathbf{Z}_{\mathbf{U}_2}, \mathbf{Z}_{\mathbf{U}_3}, \mathbf{Z}_{\boldsymbol{\mathcal{G}}}) \in \mathbb{R}^{n_1\times r_1} \times \mathbb{R}^{n_2\times r_2} \times \mathbb{R}^{n_3\times r_3} \times \mathbb{R}^{r_1\times r_2\times r_3} : \mathbf{U}_d^T\mathbf{Z}_{\mathbf{U}_d} + \mathbf{Z}_{\mathbf{U}_d}^T\mathbf{U}_d = 0,\ \text{for } d \in \{1, 2, 3\}\}$. The earlier discussion on symmetry and least-squares structure leads to the novel metric $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$

$$\begin{aligned} g_x(\xi_x, \eta_x) \quad &= \langle \xi_{\mathbf{U}_1}, \eta_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)\rangle + \langle \xi_{\mathbf{U}_2}, \eta_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)\rangle \\ &+ \langle \xi_{\mathbf{U}_3}, \eta_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)\rangle + \langle \xi_{\boldsymbol{\mathcal{G}}}, \eta_{\boldsymbol{\mathcal{G}}}\rangle, \end{aligned} \qquad (3)$$

where $\xi_x, \eta_x \in T_x\mathcal{M}$ are tangent vectors with matrix characterizations, $(\xi_{\mathbf{U}_1}, \xi_{\mathbf{U}_2}, \xi_{\mathbf{U}_3}, \xi_{\boldsymbol{\mathcal{G}}})$ and $(\eta_{\mathbf{U}_1}, \eta_{\mathbf{U}_2}, \eta_{\mathbf{U}_3}, \eta_{\boldsymbol{\mathcal{G}}})$, respectively and $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. As contrasts to the classical Euclidean metric, the metric (3) *scales* the level sets of the cost function on the search space that leads a preconditioning effect on the algorithms developed on the Tucker manifold.

3 Notions of optimization on quotient manifolds

Each point on a quotient manifold represents an entire equivalence class of matrices in the total space. Abstract geometric objects on a quotient manifold call for matrix representatives in the total space. Similarly, algorithms are run in the total space $\mathcal{M}$, but under appropriate compatibility between the Riemannian structure of $\mathcal{M}$ and the Riemannian structure of the quotient manifold $\mathcal{M}/\sim$, they define algorithms on the quotient manifold. Once we endow $\mathcal{M}/\sim$ with a Riemannian structure, the constraint optimization problem (1) is conceptually transformed into an unconstrained optimization over the Riemannian quotient manifold (2). Figure 1 illustrates a schematic view of optimization with equivalence classes, where the points x and y in $\mathcal{M}$ belong to the same equivalence class (shown in solid blue color) and they represent a single point $[x] := \{y \in \mathcal{M} : y \sim x\}$ on the quotient manifold $\mathcal{M}/\sim$. The abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ at $[x] \in \mathcal{M}/\sim$ has the matrix representation in $T_x\mathcal{M}$, but restricted to the directions that do not induce a displacement along the equivalence class $[x]$. This is realized by decomposing $T_x\mathcal{M}$ into two complementary subspaces. The vertical space $\mathcal{V}_x$ is the tangent space of the equivalence class $[x]$. On the other hand, the horizontal space $\mathcal{H}_x$ is the *orthogonal subspace* to $\mathcal{V}_x$, i.e., $T_x\mathcal{M} = \mathcal{V}_x \oplus \mathcal{H}_x$. The horizon-

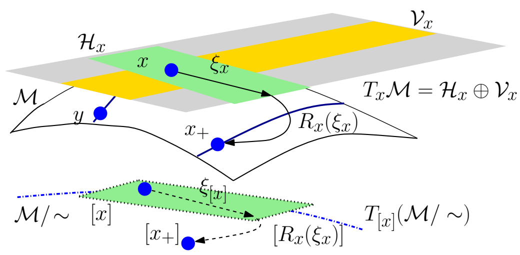


Fig. 1: Riemannian optimization framework.

Table 1: Ingredients to implement an off-the-shelf conjugate gradient algorithm for (1).

Vertical tangent vectors in $\mathcal{V}_x$	$\{(\mathbf{U}_1\mathbf{\Omega}_1, \mathbf{U}_2\mathbf{\Omega}_2, \mathbf{U}_3\mathbf{\Omega}_3, -(\boldsymbol{\mathcal{G}}\times_1\mathbf{\Omega}_1 + \boldsymbol{\mathcal{G}}\times_2\mathbf{\Omega}_2 + \boldsymbol{\mathcal{G}}\times_3\mathbf{\Omega}_3)) :$ $\mathbf{\Omega}_d \in \mathbb{R}^{r_d\times r_d}, \mathbf{\Omega}_d^T = -\mathbf{\Omega}_d, \text{for } d \in \{1,2,3\}\}$
Horizontal tangent vectors in $\mathcal{H}_x$	$\{(\zeta_{\mathbf{U}_1}, \zeta_{\mathbf{U}_2}, \zeta_{\mathbf{U}_3}, \zeta_{\boldsymbol{\mathcal{G}}}) \in T_x\mathcal{M} :$ $(\mathbf{G}_d\mathbf{G}_d^T)\zeta_{\mathbf{U}_d}^T\mathbf{U}_d + \zeta_{\mathbf{G}_d}\mathbf{G}_d^T$ is symmetric, for $d \in \{1,2,3\}\}$
$\Psi(\cdot)$ projects an ambient vector $(\mathbf{Y}_{\mathbf{U}_1}, \mathbf{Y}_{\mathbf{U}_2}, \mathbf{Y}_{\mathbf{U}_3}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}})$ onto $T_x\mathcal{M}$	$(\mathbf{Y}_{\mathbf{U}_1} - \mathbf{U}_1\mathbf{S}_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)^{-1}, \mathbf{Y}_{\mathbf{U}_2} - \mathbf{U}_2\mathbf{S}_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)^{-1},$ $\mathbf{Y}_{\mathbf{U}_3} - \mathbf{U}_3\mathbf{S}_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)^{-1}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}})$, where $\mathbf{S}_{\mathbf{U}_d}$ for $d \in \{1,2,3\}$ are solutions to $\mathbf{S}_{\mathbf{U}_d}\mathbf{G}_d\mathbf{G}_d^T + \mathbf{G}_d\mathbf{G}_d^T\mathbf{S}_{\mathbf{U}_d} = \mathbf{G}_d\mathbf{G}_d^T(\mathbf{Y}_{\mathbf{U}_d}^T\mathbf{U}_d + \mathbf{U}_d^T\mathbf{Y}_{\mathbf{U}_d})\mathbf{G}_d\mathbf{G}_d^T$
$\Pi(\cdot)$ projects a tangent vector ξ onto $\mathcal{H}_x$	$(\xi_{\mathbf{U}_1} - \mathbf{U}_1\mathbf{\Omega}_1, \xi_{\mathbf{U}_2} - \mathbf{U}_2\mathbf{\Omega}_2, \xi_{\mathbf{U}_3} - \mathbf{U}_3\mathbf{\Omega}_3,$ $\xi_{\boldsymbol{\mathcal{G}}} - (-(\boldsymbol{\mathcal{G}}\times_1\mathbf{\Omega}_1 + \boldsymbol{\mathcal{G}}\times_2\mathbf{\Omega}_2 + \boldsymbol{\mathcal{G}}\times_3\mathbf{\Omega}_3)))$, where $\mathbf{\Omega}_d$ are solutions to particular *coupled* Lyapunov equations.
$\mathrm{egrad}_x f$	$(\mathbf{S}_1(\mathbf{U}_3\otimes\mathbf{U}_2)\mathbf{G}_1^T(\mathbf{G}_1\mathbf{G}_1^T)^{-1}, \mathbf{S}_2(\mathbf{U}_3\otimes\mathbf{U}_1)\mathbf{G}_2^T(\mathbf{G}_2\mathbf{G}_2^T)^{-1},$ $\mathbf{S}_3(\mathbf{U}_2\otimes\mathbf{U}_1)\mathbf{G}_3^T(\mathbf{G}_3\mathbf{G}_3^T)^{-1}, \boldsymbol{\mathcal{S}}\times_1\mathbf{U}_1^T\times_2\mathbf{U}_2^T\times_3\mathbf{U}_3^T)\times_3\mathbf{U}_3^T),$ where $\boldsymbol{\mathcal{S}} = \frac{2}{\lvert\Omega\rvert}(\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star))$.

tal subspace provides a valid matrix representation to the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ [10, Section 3.5.8]. An abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ has a unique element $\xi_x \in \mathcal{H}_x$ that is called its *horizontal lift*. Endowed with the Riemannian metric (3), the quotient manifold $\mathcal{M}/\sim$ is a *Riemannian submersion* of $\mathcal{M}$. The submersion principle then allows to work out concrete matrix representations of abstract object on $\mathcal{M}/\sim$. Particularly, starting from an arbitrary matrix (with appropriate dimensions), two linear projections are needed: the first projection Ψ_x is onto the tangent space $T_x\mathcal{M}$, while the second projection Π_x is onto the horizontal subspace $\mathcal{H}_x$. Their formulas are shown in Table 1. The computation cost of these projections is $O(n_1r_1^2 + n_2r_2^2 + n_3r_3^2)$.

Finally, we propose a Riemannian nonlinear conjugate gradient algorithm for (1) that scales well to large-scale instances. Specifically, we use the conjugate gradient implementation of Manopt with the ingredients described in Table 1. The convergence analysis of this method follows from [15, 16, 10]. If $f(\boldsymbol{\mathcal{X}}) = \|\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star)\|_F^2/|\Omega|$, then the Riemannian gradient $\mathrm{grad}_x f$, which has the matrix characterization $\Psi(\mathrm{egrad}_x f)$, where $\mathrm{egrad}_x f$ is the Euclidean gradient of f. We show a way to compute a step-size guess effectively. The total computational cost per iteration of our proposed algorithm is $O(|\Omega|r_1r_2r_3)$, where $|\Omega|$ is the number of known entries.

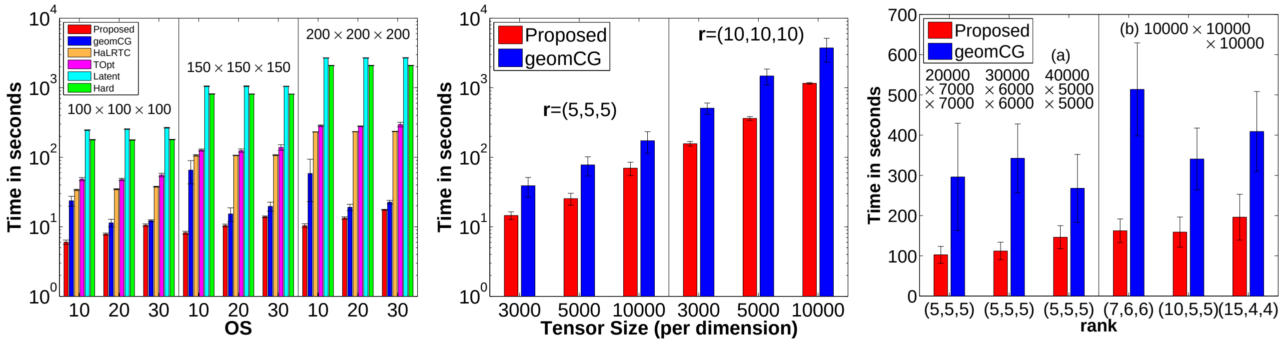
4 Numerical comparisons

We show numerical comparisons of our proposed algorithm with state-of-the-art algorithms that include TOpt [6] and geomCG [7], for comparisons with Tucker decomposition based algorithms, and HaLRTC [2], Latent [3], and Hard [4] as nuclear norm minimization algorithms. All simulations are performed in Matlab on a 2.6 GHz Intel Core i7 machine with 16 GB RAM. For specific operations with unfoldings of $\boldsymbol{\mathcal{S}}$, we use the `mex` interfaces that are provided in geomCG. For large-scale instances, our algorithm is only compared with geomCG as other algorithms cannot handle these instances. We randomly and uniformly select known entries based on a multiple of the dimension, called the *over-sampling* (OS) ratio, to create the training set Ω. Algorithms (and problem instances) are initialized randomly, as in [7], and are stopped when either the mean square error (MSE) on the training set Ω is below 10^{-12} or the number of iterations exceeds 250. We also evaluate the mean square error on a test set Γ, which is different from Ω. Five runs are performed in each scenario.

Case 1 considers synthetic small-scale tensors of size $100\times100\times100$, $150\times150\times150$, and $200\times200\times200$ and rank $\mathbf{r} = (10, 10, 10)$ are considered. OS is $\{10, 20, 30\}$. Figure 2(a) shows that the convergence behavior of our proposed algorithm is either competitive or faster than the others. Next, **Case 2** considers large-scale tensors of size $3000 \times 3000 \times 3000$, $5000 \times 5000 \times 5000$, and $10000 \times 10000 \times 10000$ and ranks $\mathbf{r} = (5, 5, 5)$ and $(10, 10, 10)$. OS is 10. Our proposed algorithm outperforms geomCG in Figure 2(b). **Case 3** considers instances where the dimensions and ranks along certain modes are different than others. Two cases are considered. Case (3.a) considers tensors size $20000 \times 7000 \times 7000$, $30000 \times 6000 \times 6000$, and $40000 \times 5000 \times 5000$ with rank $\mathbf{r} = (5, 5, 5)$. Case (3.b) considers a tensor of size $10000 \times 10000 \times 10000$ with ranks $(7, 6, 6)$, $(10, 5, 5)$, and $(15, 4, 4)$. In all the cases, the proposed algorithm converges faster than geomCG as shown in Figure 2(c). Finally, **Case 4** considers MovieLens-10M dataset that contains 10000054 ratings corresponding to 71567 users and 10681 movies. We split the time into 7-days wide bins results, and finally, get a tensor of size $71567 \times 10681 \times 731$. The fraction of known entries is less than 0.002%. We perform five random 80/10/10–train/validation/test partitions. The maximum iteration is set to 500. As shown in Table 2, our proposed algorithm consistently gives lower test errors than geomCG across different ranks.

5 Conclusion and future work

We have proposed a preconditioned nonlinear conjugate gradient algorithm for the tensor completion



(a) **Case 1:** small-scale tensors.

(b) **Case 2:** large-scale tensors.

(c) **Case 3:** asymmetric tensors.

Fig. 2: Experiments on synthetic datasets.

Table 2: **Case 4:** test MSE on Γ and time in seconds.

MovieLens-10M	Proposed		geomCG	
r	Time	MSE on Γ	Time	MSE on Γ
$(4,4,4)$	$\mathbf{1748 \pm 441}$	$\mathbf{0.6762 \pm 1.5 \cdot 10^{-3}}$	2981 ± 40	$0.6956 \pm 2.8 \cdot 10^{-3}$
$(6,6,6)$	$\mathbf{6058 \pm 47}$	$\mathbf{0.6913 \pm 3.3 \cdot 10^{-3}}$	6554 ± 655	$0.7398 \pm 7.1 \cdot 10^{-3}$
$(8,8,8)$	$\mathbf{11370 \pm 103}$	$\mathbf{0.7589 \pm 7.1 \cdot 10^{-3}}$	13853 ± 118	$0.8955 \pm 3.3 \cdot 10^{-2}$
$(10,10,10)$	$\mathbf{32802 \pm 52}$	$\mathbf{1.0107 \pm 2.7 \cdot 10^{-2}}$	38145 ± 36	$1.6550 \pm 8.7 \cdot 10^{-2}$

problem by exploiting the fundamental structures of symmetry, due to non-uniqueness of Tucker decomposition, and least-squares of the cost function. A novel Riemannian metric is proposed that enables to use the versatile Riemannian optimization framework. Numerical comparisons suggest that our proposed algorithm has a superior performance on different benchmarks.

Acknowledgement

Hiroyuki Kasai is (partly) supported by the Ministry of Internal Affairs and Communications, Japan, as the SCOPE Project (150201002). Bamdev Mishra was supported as an FNRS research fellow (Belgian Fund for Scientific Research). The scientific responsibility rests with its authors.

参考文献

1) E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.
2) J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):208–220, 2013.
3) R. Tomioka, K. Hayashi, and H. Kashima. Estimation of low-rank tensors via convex optimization. Technical report, arXiv:1010.0789, 2011.
4) M. Signoretto, Q. T. Dinh, L. D. Lathauwer, and J. A. K. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Mach. Learn.*, 94(3):303–351, 2014.
5) T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.
6) M. Filipović and A. Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidim. Syst. Sign. P.*, 2013. Doi: 10.1007/s11045-013-0269-9.
7) D. Kressner, M. Steinlechner, and B. Vandereycken. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.*, 54(2):447–468, 2014.
8) J. Nocedal and S. J. Wright. *Numerical Optimization*, volume Second Edition. Springer, 2006.
9) B. Mishra and R. Sepulchre. Riemannian preconditioning. Technical report, arXiv:1405.6055, 2014.
10) P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.
11) S. T. Smith. Optimization techniques on Riemannian manifold. In A. Bloch, editor, *Hamiltonian and Gradient Flows, Algorithms and Control*, volume 3, pages 113–136. Amer. Math. Soc., Providence, RI, 1994.
12) A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.
13) N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt: a Matlab toolbox for optimization on manifolds. *JMLR*, 15(1):1455–1459, 2014.
14) J. M. Lee. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2003.
15) H. Sato and T. Iwai. A new, globally convergent Riemannian con ugate gradient method. *Optimization*, 64(4):1011–1031, 2015.
16) W. Ring and B. Wirth. Optimization methods on riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012.

量子回路対応ニューラルネットワークの時系列予測問題への適応

○上口大晴 松井伸之 礒川悌次郎 （兵庫県立大学）

An Application of uantum MLP with Complex-Valued Representation to Chaotic Time Series Prediction Problem

∗T. Ueguchi, N. Matsui and T. Isokawa (University of Hyogo)

概要 量子回路対応ニューラルネットワーク (QNN) は量子計算を導入したニューラルネットワークである．QNN は量子ビットニューロンモデルから構成されており，その内部状態は量子ビットの重ね合わせ状態により表現される．本研究では，カオス系列の一つであるローレンツアトラクタの時系列予測問題を通して QNN の性能評価を行い，通常の NN と比較することで QNN の有用性を示す．実験結果より，QNN は NN より高い精度で予測することが可能であることが示された．

キーワード: 量子回路対応ニューラルネットワーク, 時系列予測

1 はじめに

近年，Google の Deep learning 研究の成果 [1)] などに端を発し，また Big データ時代を反映してニューラルネットワークなどをはじめとする様々な計算知能 (Computational Intelligence) 研究が勢いづいている．それのみならず，量子コンピュータの実用性が高まるにつれ，量子情報技術の実現性が高り，これらの計算知能を量子情報科学分野へ拡張融合し，これまでにない情報処理性能の創出をめざした研究，すなわち量子計算知能に関する基礎および応用研究がますます進展している [2)]．現在では，量子ニューラルネットワーク (QNN) や量子進化的アルゴリズム (QEA) 並びに量子粒子群最適化法 (QPSO) など数多くの量子計算知能研究例が文献に見られる [3)]．これらは伝統的な計算手法では解くことが困難な問題を解きうる手法として今後ますます期待される．そこで，本論文では，このような研究動向に鑑みて，カオス時系列データ予測問題を困難な問題の例として取り上げ，その予測手法としての量子ニューラルネットワークの性能を評価する．

ニューラルネットワークによる時系列予測は以前から数多く研究されその有用性が示されてきた [4)]．近年では，株価予測問題 [5)] や津波予測問題 [6)] など，カオス的な時系列問題も議論されている [7)]．しかし，カオスデータは初期値に敏感で，長期の予測が困難であり，実用的な予測を行いうるためにはその精度の向上が不可欠である．そのような観点から，量子ニューラルネットワークの例として量子回路対応ニューラルネットワーク [8)] を用い，従来のニューラルネットワークとの性能比較を行って性能向上手法としての量子計算知能の有用性を探る．調査したモデルは，具体的な実用問題解法のために複素数表示を用いて量子計算をニューラルネットワークに具現化し得た，著者らが提唱した初期モデル [9)] であるが，このモデルと通常のニューラルネットワークの性能差をローレンツアトラクタの予測を通じて再吟味した結果をここでは報告する．

2 量子計算

QNN を構成する量子ビットニューロンモデルは，その内部状態を量子ビットを用いて記述し，内部状態の計算を量子計算で行う．このニューロンモデルにおいて，量子計算は量子論理ゲートを用いて行われる．

2.1 量子ビット

量子計算では, 量子ビット (qubit) と呼ばれるビット表現を用いて計算を行う．量子ビットとは，従来的なビット表現の’0’，’1’ に量子力学的解釈を取り入れたビット表現で，$|0\rangle$，$|1\rangle$ と表現される．この量子ビット表現が従来的なビット表現と異なるのは，ビットの状態を量子状態として扱い，その重ね合わせで表現することである．任意の量子ビット状態 $|\phi\rangle$ は，

$$|\phi\rangle = a\,|0\rangle + b\,|1\rangle \tag{1}$$

と表される．ここで a，b は確率振幅と呼ばれる複素数であり，任意の量子ビット状態 $|\phi\rangle$ を観測したときそれぞれの状態 $|0\rangle$，$|1\rangle$ が観測される確率を示す係数である．各状態が観測される確率は量子力学の確率解釈に従い，各状態における確率振幅における確率振幅の絶対値の平方で与えられる．よって a，b について以下の関係が成り立つ．

$$|a|^2 + |b|^2 = 1 \tag{2}$$

式 (1)，(2) より，任意の量子ビット状態 $|\phi\rangle$ の幾何学的表現は次式で表される．

$$|\phi\rangle = \cos\theta\,|0\rangle + e^{i\psi}\sin\theta\,|1\rangle \tag{3}$$

式 (3) は Bloch 球と呼ばれる球面によって表現することができる．

ここで，式 (3) で表現された量子状態を $e^{i\psi} = 1$ とし，$|0\rangle$ の確率振幅 a を実部に $|1\rangle$ の確率振幅 b を虚部に対応させた複素関数表示を用いて表現する．複素関数表示を用いて表現された量子状態は,

$$f(\theta) = e^{i\theta} = \cos\theta + i\sin\theta \tag{4}$$

と表される．ここで i は虚数単位である．

2.2 量子論理ゲート

量子回路で計算を行うために，量子論理ゲートが必要となる．この量子論理ゲートを構成する基本量子ゲートには，1 ビット回転ゲートと 2 ビット制御 NOT ゲートがあり，この 2 つのゲートを組み合わせることによって任意の量子回路を実現する．

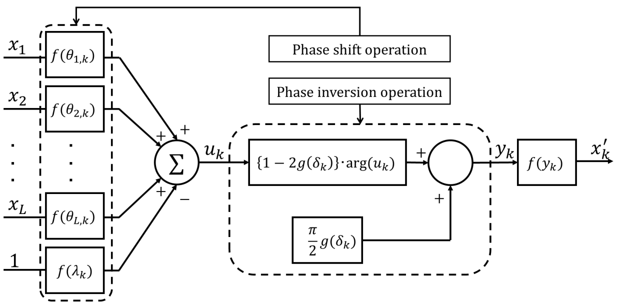


Fig. 1: Qubit neuron model

2.2.1　1 ビット回転ゲート

1 ビット回転ゲートは量子状態の位相をシフトする位相変換ゲートである．式 (4) の表現から以下に示す積形式でこのゲートを表現する．

$$f(\theta_1+\theta_2)=f(\theta_1)\cdot f(\theta_2) \tag{5}$$

2.2.2　2 ビット制御 NOT ゲート

2 ビット制御 NOT を記述するためには，量子状態の反転・無反転を表現しなければならない．そこで，反転制御入力パラメータ γ を導入することで以下に示す表現で実現する．

$$\begin{aligned} y &= f\Big(\frac{\pi}{2}\gamma+(1-2\gamma)\cdot\theta\Big) \\ &= \begin{cases} \cos\theta+i\sin\theta & (\gamma=0) \\ \sin\theta+i\cos\theta & (\gamma=1) \end{cases} \end{aligned} \tag{6}$$

ここで式 (6) において，$\gamma=1$ が反転に対応し，$\gamma=0$ が無反転に対応している [8]．

3　量子回路対応ニューラルネットワーク

量子回路対応ニューラルネットワーク (QNN) は量子力学的効果を導入した NN である．QNN を構成する量子ビットニューロンモデルは，2 章で述べた量子回路を用いて内部状態を計算し，その状態を複素関表示を用いた量子位相として表現する．このモデルにおいて，ニューロンの状態はニューロンの発火，非発火の量子重ね合わせ状態として扱うことができる．

3.1　量子ビットニューロンモデル

Fig.1 に量子状態を式 (4) の複素関数表現を用いて表現した量子ビットニューロンモデルを示す．このモデルは従来のニューロンモデルの発火，非発火を量子ビットの状態 $|1\rangle$，$|0\rangle$ に対応させたモデルである．

このモデルにおいて L 個のニューロンからの入力を受ける k 番目のニューロン出力 x'_k は式 (4) より

$$x'_k=f(y_k)=\cos y_k+i\sin y_k=e^{iy_k} \tag{7}$$

と与えられる．また，ニューロンの内部状態 u_k と k 番目のニューロンの位相入力 y_k は

$$u_k = \sum_{l=1}^{L} e^{i\theta_{l,k}}\cdot x_l - e^{i\lambda_k} \tag{8}$$

$$y_k = \frac{\pi}{2}\cdot g(\delta_k)+\{1-2g(\delta_k)\}\cdot\arg(u_k) \tag{9}$$

となる．ここで，$\theta_{l,k}$ と λ_k はそれぞれ従来の NN の結合荷重としきい値に対応する位相パラメータ，δ_k は反転度パラメータと呼ばれる位相反転の度合いを示すパラメータである．また，$g(\cdot)$ はシグモイド関数である．

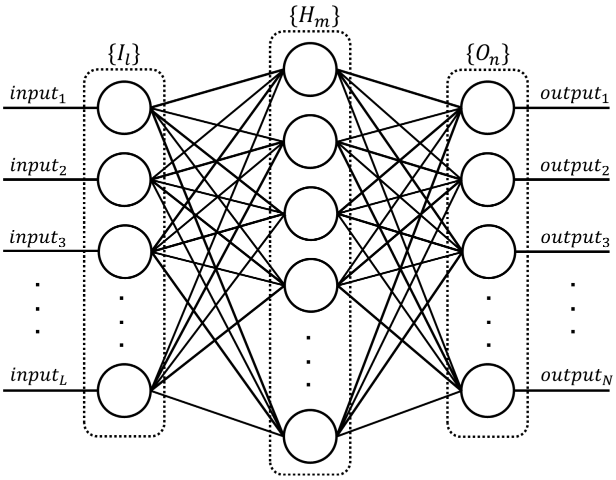


Fig. 2: 3-layerd network

以上で与えた式 (8)，(9) は，それぞれ基本量子ゲートに対応している．式 (8) は 1 ビット回転ゲートに対応し，従来のニューロンモデルの特徴である空間加算性を位相回転によって表現している．また，式 (9) は 2 ビット制御 NOT ゲートに対応し，式 (6) における反転制御パラメータ γ を可変にした表現となる [10]．

3.2　ネットワーク構造

Fig.1 の量子ビットニューロンモデルを用いて，Fig.2 に示す入力層 $\{I_l\}$ $(l=1,2,\cdots,L)$，中間層 $\{H_m\}$ $(m=1,2,\cdots,M)$，出力層 $\{O_n\}$ $(n=1,2,\cdots,N)$ からなる 3 層の階層型 NN を構成し，この NN を本研究で扱う QNN とする．以下，各層のニューロン数 L，M，N を用いて 3 層構造の NN を L-M-$N(D_n)$ と表現する．D_n はニューロンに関するパラメータの自由度である．

QNN では，量子位相表現を用いて計算を行うため，入力値を量子位相表現に変換しなければならない．よって，入力層 I の l 番目のニューロンに入力される入力値 $input_l$ は以下に示す式で位相値 y^I_l に変換される．

$$y^I_l=\frac{\pi}{2}\cdot input_l \tag{10}$$

ここで，$input_l$ は $[0,1]$ の範囲の実数であり，入力値は $[0,\pi/2]$ の範囲の位相をもつ量子状態に変換される．また，QNN の出力層 O の n 番目から得られる出力値 $output_n$ は，量子力学の確率解釈に基づき，そのニューロンの発火状態 $|1\rangle$ が観測される確率によって，

$$output_n=|\mathrm{Im}(x^O_n)|^2 \tag{11}$$

と与えられる．ここで，x^O_n は出力層 O の n 番目のニューロンの出力である．

パラメータの学習アルゴリズムには，誤差逆伝播法 (BP) を使用し，評価関数として 2 乗誤差関数を用いる．

2 乗誤差関数は次式で与えられる．

$$E_{total} = \frac{1}{2}\sum_{p}^{K}\sum_{n}^{N}(t_n^p - output_{n,p})^2 \tag{12}$$

ここで，K は学習パターンの総数，$output_{n,p}$ は p 番目の入力パターンにおける n 番目のニューロンから出力されるネットワーク出力，t_n^p はその出力に対応する教師信号である．

パラメータの更新には最急降下法を用いる．パラメータの更新式は，

$$\theta_{l,k}^{new} = \theta_{l,k}^{old} - \eta\frac{\partial E_{total}}{\partial \theta_{l,k}^{old}} \tag{13}$$

$$\lambda_{k}^{new} = \lambda_{k}^{old} - \eta\frac{\partial E_{total}}{\partial \lambda_{k}^{old}} \tag{14}$$

$$\delta_{k}^{new} = \delta_{k}^{old} - \eta\frac{\partial E_{total}}{\partial \delta_{k}^{old}} \tag{15}$$

と与えられる．ここで，η は学習係数と呼ばれるパラメータである [10)]．

4 実験

本章では，カオス系列の一つであるローレンツアトラクタの時系列問題を通して QNN の性能評価を行い，通常の NN をと比較して QNN の有用性を示す．

4.1 ローレンツアトラクタ

ローレンツアトラクタはローレンツ方程式に基づいて，状態が遷移するカオス系列モデルである．このローレンツ方程式は次に示す 3 つの微分方程式から構成され，

$$\frac{dx}{dt} = \sigma(y-x) \tag{16}$$

$$\frac{dy}{dt} = x(\rho - z) - y \tag{17}$$

$$\frac{dz}{dt} = xy - \beta z \tag{18}$$

と表現される．ここで，x，y，z は時刻 t におけるローレンツアトラクタの状態を表す．また，σ，ρ，β はローレンツ方程式のパラメータである．

このローレンツ方程式のパラメータを特定の値に設定することによって，この系はカオス系列になる．例えば，ローレンツ方程式のパラメータを $\sigma = 10$，$\rho = 28$，$\beta = 8/3$ に設定すと，ローレンツ方程式から得られる系は 2 つのアトラクタを有するカオス系列になる．

4.2 実験方法

ローレンツアトラクタの状態を予測するネットワークを，QNN，NN それぞれについて table1 に示すように構成する．このとき，QNN と NN のパラメータの自由度 D_n を同程度になるように設定する．それぞれ設定したネットワークにおいて，時刻 $t-2$，$t-1$，t における状態 x，y，z を入力したときに，時刻 $t+1$ の状態が出力されるように学習を行う．

本実験において，ローレンツ方程式のパラメータ σ，ρ，β を 4.1 節で示した値に設定し，ローレンツアトラクタの初期状態を $x = 1.0$，$y = 1.0$，$z = 1.0$ とする．本実験に用いるデータは，時刻 $t \in [1, 30]$ の範囲を微小時間 $\Delta t = 0.01$ 間隔で区切った，3000 ステップ分の状態とする．この状態を生成するにあたって，式 (16)，(17)，(18) で示したローレンツ方程式をオイラー法を解くことにより求めている．以上で得られたデータを $t \in [1, 20]$ の範囲の 2000 ステップ分を学習データ，それ以外の 1000 ステップ分をテストデータに分割する．また，ネットワークへの入力は，$[0, 1]$ の範囲の実数値でないとならないので，正規化を行ってからネットワークに入力する．

Table 1: Structure of neural networks

	QNN	NN
L	9	9
M	45	49
N	3	3
D_n	636	640

Table 2: Learning results($ER = 0.1$)

	QNN	NN
Max success rate(%)	100	100
Learning iteration	558	10660

4.3 実験結果

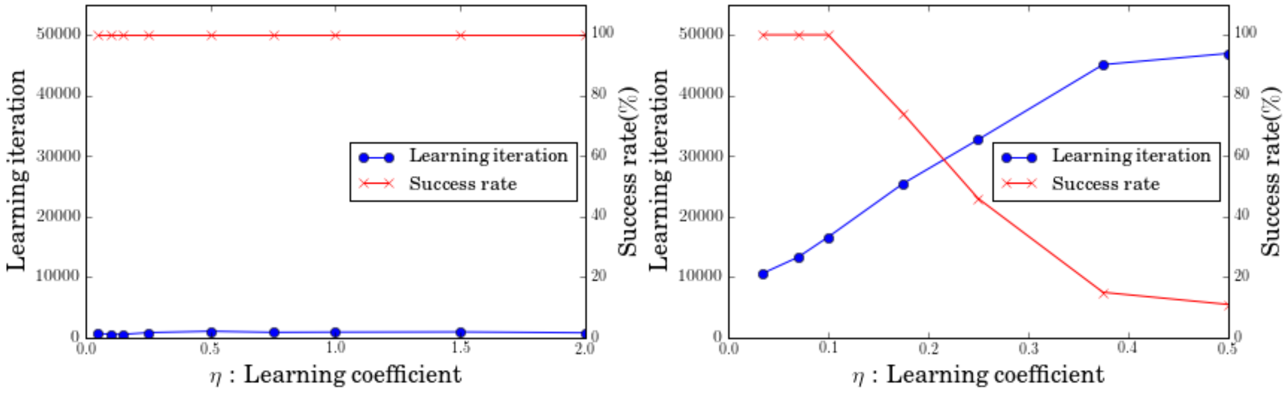
まず，QNN と NN の打ち切り誤差 ER を同条件に設定して実験を行った．本実験におけるニューラルネットワークの学習の収束条件は式 (12) に示した評価関数の値が ER を下回ったときとする．また，最大学習回数 LM を設定する．学習が設定した LM に達しても，評価関数の値が ER を下回らなかった場合，学習が収束しなかったとみなす．ここで学習 1 回とは，学習データをネットワークにすべて入力し終えたときとする．

$ER = 0.1$，$LM = 50000$ に設定し，学習係数 η を変化させながら，100 回試行時の平均学習回数と学習成功率を調査する．この試行において，学習が収束しなかった場合の学習回数は LM としている．実験結果を Fig.3 に示す．また，table2 に最大学習成功率とそのときの平均学習回数を示す．ただし，table2 に示す平均学習回数は最大学習成功率の中で最小の値を示している．table2 より，QNN は NN に比べて非常に少ない学習回数で学習が収束していることがわかる．また Fig.3 より，QNN は η の値に関わらず，学習成功率 100% で学習に成功しているのに対して，NN は $\eta \in [0.035, 0.1]$ の範囲で，学習成功率 100% で学習に成功していることがわかる．

Fig.4 に，この場合のローレンツアトラクタの軌跡の出力結果を示しておく．Fig.4 から QNN，NN どちらもローレンツアトラクタの軌跡を追えていることがわかる．しかし，精度をさらに 1/20 に上げて $ER = 0.005$ とした場合，QNN では 28929 回で収束するが，NN では学習係数 η をどのように設定しても最大学習制限の 50000 回内では収束できないことがわかった．Fig.5 に示したように，QNN がローレンツアトラクタ軌跡をこの場合にも精度よく追従できていることがわかる．

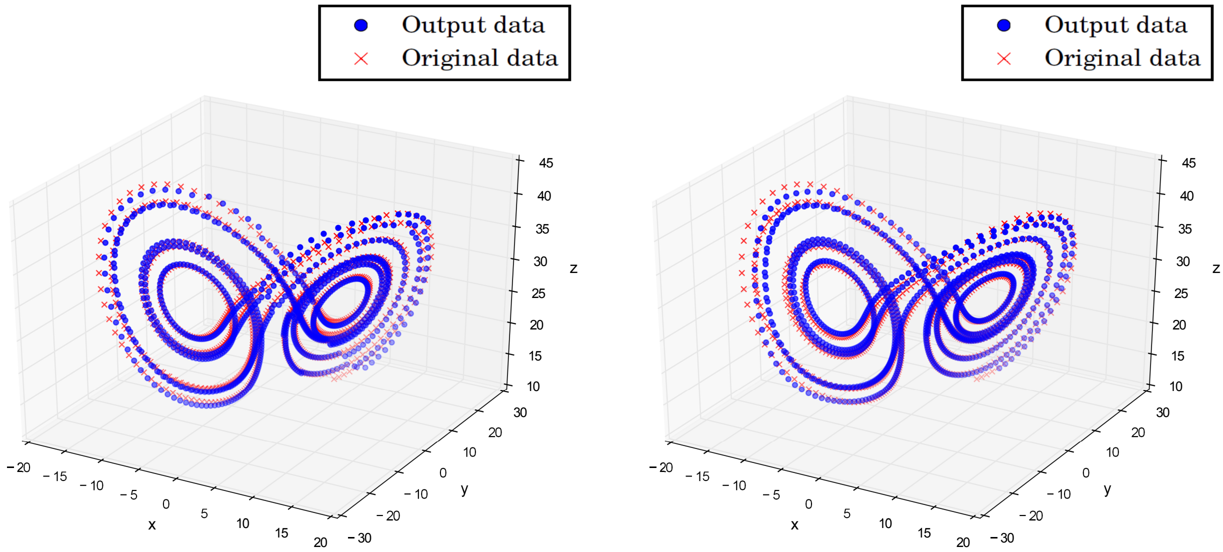
5 まとめ

本論文では，ローレンツアトラクタの時系列問題を通して QNN の性能を NN と比較することによって評



(a) QNN (b) NN

Fig. 3: Learning coefficient-dependence of the learning iteration and the learning success rate



(a) Lorenz Attractor(QNN : $\eta = 0.05$) (b) Lorenz Attractor(NN : $\eta = 0.035$)

Fig. 4: Output Results($ER = 0.1$)

価した．その結果，QNN は NN に比べ，高速に学習が収束し，ローレンツアトラクタの軌跡を忠実に再現できることが示された．以上のことにより，QNN はカオス系列であるローレンツアトラクタの時系列問題に対して有用であることがわかった．

今後の課題としては，ローレンツアトラクタ以外のカオス系列の時系列問題に対する QNN の有用性の調査，および量子ビットニューロンの構造の改良，およびそれに対応した学習アルゴリズムの開発などが挙げられる．

参考文献

1) Quoc V. Le et al. “Building High-level Features Using Large Scale Unsupervised Learning”. *Proc.the 29th Int. Conf. on Machine Learning*, pp. 8595–8598, (2012).

2) 松井伸之. “量子計算知能展望”. 計測と制御, Vol. 54, No. 8, pp. 579–585, (2015).

3) A. Man u and M. J. Nigam. “Application of quantum inspired computational intelligence: a survey”. *Artificial Intelligence Review*, Vol. 42, No. 1, pp. 79–156, (2014).

4) Y. M. Hu G. Zhang, B. E. Patuwo. “Forecasting with artificial neural networks: The state of the art”. *International Journal of Forecasting*, Vol. 14, No. 1, pp. 35–62, (1998).

5) 馬場則夫. “ニューラルネットの株価予測への適応”. 白鳥則郎他（編）, “ソフトコンピューティングの基礎と応用”, pp. 64–85. 共立出版, (2012).

6) 間瀬 肇他. “ニューラルネットワークを用いた大阪湾内のリアルタイム津波予測”. 京都大学防災研究所年報, Vol. 50 B, pp. 527–535, (2007).

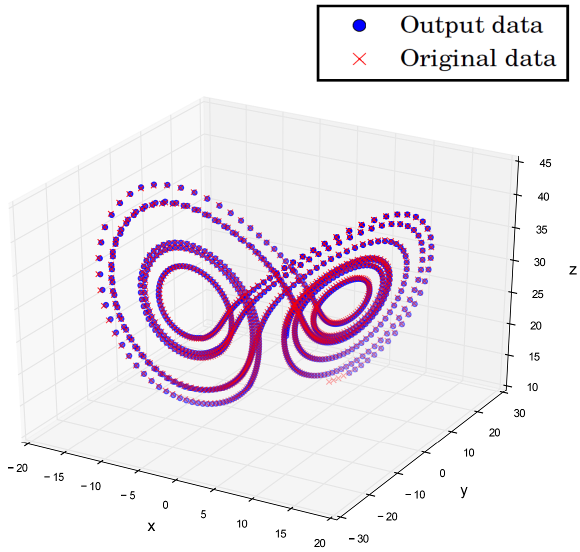


Fig. 5: Lorenz Attractor(QNN : $\eta = 0.05$, $ER = 0.005$)

7) S. V. Dudul. "Prediction of a Lorentz chaotic attractor using two-layer perceptoron neural network". *Applied Soft Computing*, Vol. 5, No. 4, pp. 333–355, (2005).

8) 松井伸之, 幸田憲明, 西村治彦. "量子描像ニューラルネットワーク". 渡辺桂吾（編）, "ニューラルネットワーク計算知能", pp. 90–113. 森北出版, (2006).

9) 松井伸之, 高井真人, 西村治彦. "量子描像ニューロンに基づく量子回路対応ネットワークモデル". 電子情報通信学会論文誌. A, 基礎・境界, Vol. 81, No. 12, pp. 1687–1692, (1998).

10) N. Kouda, N. Matsui, and H. Nishimura. "A multilayered feed-forward network based on qubit neuron model". *Syst. Comput. Japan*, Vol. 35, No. 13, pp. 43–51, (2004).

四元数２部グラフ自己連想記憶の想起性能評価

○峯本俊文[1] 礒川悌次郎[1] 小林正樹[2] 西村治彦[1] 松井伸之[1]
([1]兵庫県立大学，[2]山梨大学)

Performance Analysis for Quaternionic Bipartite Auto-Associative Memory

*T. Minemoto[1], T. Isokawa[1], M. Kobayashi[2], H. Nishimura[1] and N. Matsui[1]
([1]University of Hyogo, [2]University of Yamanashi)

Abstract– Quaternionic Bipartite Auto-Associative Memory (QBAAM) is an associative memory neural network which can store and retrieve patterns with multiple levels. A part of neurons in the network are quaternionic neurons. These neurons can represent three kinds of discretized phases, i.e., three-dimensional multi-level values. The rest of neurons are conventional (real-valued) neurons. We show that QBAAM has better noise robustness than conventional Quaterninic Hopfield Associative Memory (QHAM) through numerical simulations.

Key Words: Associative Memory, Quaternion

1 はじめに

近年，複素数に基づいたニューラルネットワーク(NN) に関する研究が広く行われている [1, 2]．また，複素数よりもさらに高次元の数である四元数を NN に導入する試みも進められている．四元数は 4 成分からなる超複素数であり，3 次元空間における幾何学変換を簡潔に表現できることから，コンピュータグラフィクスや姿勢制御などに応用されている [3]．四元数を導入した NN は，従来型の NN と比較して，3 次元空間での情報処理などにおいて効率的な処理の実現が期待できる．これまでに，剛体制御 [4]，カラーナイトビジョン [5]，風向予測 [6, 7] などの工学問題への応用が試みられている．

NN を用いた連想記憶に関する研究は，Hopfield 連想記憶 (Hopfield Associative Memory: HAM) を始めとして，数多くの研究が行われている [8]．四元数 Hopfield 連想記憶 (Quaternionic Hopfield Associative Memory: QHAM) はニューロンの入出力，結合荷重，内部状態の表現に四元数を用いた Hopfield 型の連想記憶モデルである [9]．振幅および 3 種類の位相成分によって四元数を表現することで [10]，3 成分の多値パターンの記銘・想起が可能である [11]．文献 [9] では，四元数に拡張した射影学習則により，カラー画像の学習が可能であることを示した．しかしながら，QHAM は高い表現力をもつ一方，学習パターンの回転不変性に起因する偽記憶が多く，雑音耐性が低いことがわかっている．この問題を解決する一手法として，四元数 2 部グラフ連想記憶 (Quaternionic Bipartite Auto-Associative Memory: QBAAM) が提案された [12]．QBAAM は四元数ニューロンで構成される可視層と実数ニューロンで構成される不可視層の 2 層のネットワーク構造をもつ連想記憶モデルである．複素数を用いたモデルでは，先に述べた偽記憶の削減に成功し，雑音耐性が向上することが数値実験により示されている [13]．QBAAM においても，偽記憶の削減とそれに伴う雑音耐性の改善が期待される．

本稿では，QHAM と QBAAM における学習パターンと主たる偽記憶である回転パターンの安定性を評価することで，QBAAM による偽記憶の削減効果を調べる．また，QBAAM が QHAM と比較して高い雑音耐性をもつことを数値実験により示す．

2 準備

2.1 四元数

四元数は 1 つの実数と 3 つの虚数からなる超複素数である．3 つの虚数単位を $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ とすると，四元数 x は，

$$x = x^{(e)} + x^{(i)}\boldsymbol{i} + x^{(j)}\boldsymbol{j} + x^{(k)}\boldsymbol{k} \tag{1}$$

と表される．ここで $x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}$ は実数であり，それぞれ四元数 x の各成分を表す．四元数の虚数単位においては，次式で示される Hamilton 関係が成立する．

$$\begin{aligned} &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \\ &\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k},\ \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i},\ \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}. \end{aligned} \tag{2}$$

$\boldsymbol{ij} \neq \boldsymbol{ji}$ で示されるとおり，四元数は非可換の数体系である．また，四元数は，スカラ部分 $x^{(e)}$ とベクトル部分 $\vec{x} = (x^{(i)}, x^{(j)}, x^{(k)})$ に分けることにより，

$$x = (x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}) = (x^{(e)}, \vec{x}) \tag{3}$$

とも表記される．四元数 x の共役な四元数 x^* は

$$\begin{aligned} x^* &= (x^{(e)}, -\vec{x}) \\ &= x^{(e)} - x^{(i)}\boldsymbol{i} - x^{(j)}\boldsymbol{j} - x^{(k)}\boldsymbol{k} \end{aligned} \tag{4}$$

と定義される．以上の定義を用いると，四元数 p, q の加減算 $p \pm q$ と積 pq は以下のようになる．

$$\begin{aligned} p \pm q &= (p^{(e)} \pm q^{(e)}, \vec{p} \pm \vec{q}) \\ &= (p^{(e)} \pm q^{(e)}, p^{(i)} \pm q^{(i)}, p^{(j)} \pm q^{(j)}, p^{(k)} \pm q^{(k)}), \end{aligned} \tag{5}$$

$$pq = (p^{(e)}q^{(e)} - \vec{p}\cdot\vec{q}, p^{(e)}\vec{q} + q^{(e)}\vec{p} + \vec{p}\times\vec{q}). \tag{6}$$

ここで，$\vec{p}\cdot\vec{q}$ と $\vec{p}\times\vec{q}$ はそれぞれ 3 次元ベクトル $\vec{p},\vec{q}$ の内積と外積である．

2.2　極座標表現に基づく四元数ニューロン

四元数 x は，振幅 $|x|$ と 3 種類の位相変数 φ, θ, ψ を用いることで極座標表現が可能である [10)]．四元数 x は，

$$x = |x| e^{\boldsymbol{i}\varphi} e^{\boldsymbol{k}\psi} e^{\boldsymbol{j}\theta} \tag{7}$$

と表され，ここで，

$$e^{\boldsymbol{i}\varphi} = \cos\varphi + \boldsymbol{i}\sin\varphi \quad (-\pi \leq \varphi < \pi), \tag{8}$$

$$e^{\boldsymbol{j}\theta} = \cos\theta + \boldsymbol{j}\sin\theta \quad (-\pi/2 \leq \theta < \pi/2), \tag{9}$$

$$e^{\boldsymbol{k}\psi} = \cos\psi + \boldsymbol{k}\sin\psi \quad (-\pi/4 \leq \psi \leq \pi/4) \tag{10}$$

である．

極座標表現の四元数の各位相を離散化することにより，四元数に基づいた多値ニューロンを構成できる．3 つの位相 φ, θ, ψ の量子化単位をそれぞれ $\varphi_0 = 2\pi/A$, $\theta_0 = \pi/B$, $\psi_0 = \pi/2C$ とすると，四元数ニューロンの状態 $q_{a,b,c}(a \in \{0,1,\ldots,A-1\}, b \in \{0,1,\ldots,B-1\}, c \in \{0,1,\ldots,C-1\})$ は次式により定義できる．

$$q_{a,b,c} = q_a^{(\varphi)} q_c^{(\psi)} q_b^{(\theta)}. \tag{11}$$

ここで，

$$q_a^{(\varphi)} = \exp\left(\boldsymbol{i}(-\pi + a\varphi_0 + \varphi_0/2)\right), \tag{12}$$

$$q_b^{(\theta)} = \exp\left(\boldsymbol{j}(-\pi/2 + b\theta_0 + \theta_0/2)\right), \tag{13}$$

$$q_c^{(\psi)} = \exp\left(\boldsymbol{k}(-\pi/4 + c\psi_0 + \psi_0/2)\right) \tag{14}$$

である．つまり，1 つの四元数ニューロンは $A \times B \times C$ 通りの状態を取ることができる．$A = 4, B = 3, C = 2$ とした場合の例を Fig. 1 に示す．この場合，1 つの四元数ニューロンは 24 通りの状態をとる．

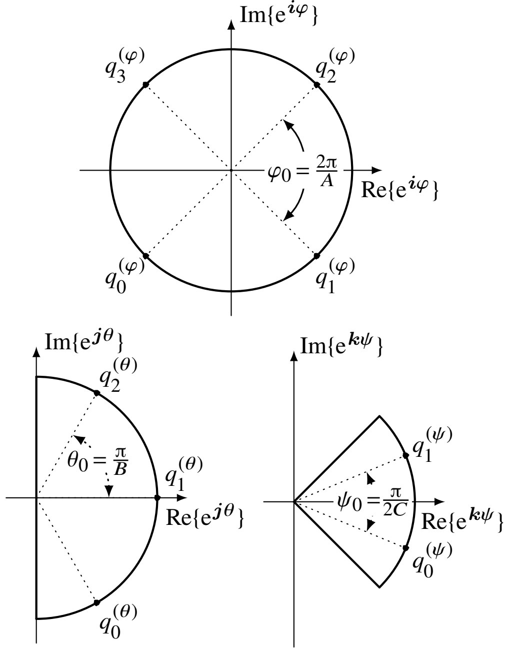


Fig. 1: States of a quaternionic neuron ($A = 4$, $B = 3$, $C = 2$)

3　四元数 2 部グラフ自己連想記憶

QBAAM は，Fig. 2 に示すように，四元数ニューロンで構成される可視層と実数ニューロンで構成される不可視層の 2 層のネットワーク構造をもつ連想記憶モデルである．各ニューロンは層間で相互に結合しており，層内におけるニューロンの結合はない．

3.1　想起

可視層と不可視層の状態ベクトルをそれぞれ $\mathbf{x} = (x_1, x_2, \ldots, x_N)^{\mathrm{T}}$, $\mathbf{y} = (y_1, y_2, \ldots, y_M)^{\mathrm{T}}$ とすると，時刻 t における各層のニューロンの状態は次式のように定義される．

$$y_i(t) = f\left(\sum_{j=1}^{N} w_{ij} x_j(t)\right), \tag{15}$$

$$x_i(t+1) = g\left(\sum_{j=1}^{M} v_{ij} y_j(t)\right). \tag{16}$$

ここで，N, M はそれぞれ可視層，不可視層におけるニューロンの数である．また，$w_{ij} \in \mathbb{H}$ は可視層の j 番目のニューロンから不可視層の i 番目のニューロンへの結合荷重，$v_{ij} \in \mathbb{H}$ は不可視層の j 番目のニューロンから可視層の i 番目のニューロンへの結合荷重である．

式 (15), (16) において，$f(\cdot)$ および $g(\cdot)$ は，それぞれ不可視層と可視層のニューロンの活性化関数である．QBAAM では，次式で定義される符号関数を不可視層の活性化関数として用いる．

$$f(x) = \begin{cases} 1 & x^{(e)} \geq 0 \\ -1 & x^{(e)} < 0 \end{cases}. \tag{17}$$

つまり，不可視層のニューロン出力は 1 か −1 のどちらかの状態をとる．可視層の活性化関数には，複素多値 NN で用いられる活性化関数 [14)] を四元数へ拡張したものを用いる．その定義は以下のとおりである．

$$g(x) = \mathrm{qsign}(x) \tag{18}$$

$$= \mathrm{csign}_A(x)\,\mathrm{csign}_C(x)\,\mathrm{csign}_B(x). \tag{19}$$

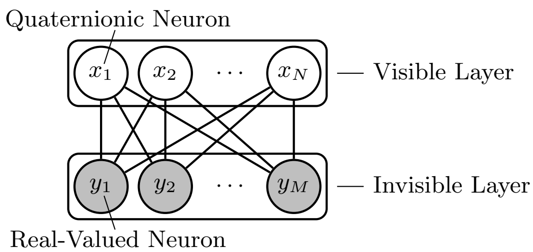


Fig. 2: The architecture of quaternionic bibdirectional auto-associative memory (QBAAM).

ここで，

$$\mathrm{csign}_A(x) = \begin{cases} q_0^{(\varphi)} & -\pi \le \arg_\varphi(x) < -\pi + \varphi_0 \\ q_1^{(\varphi)} & -\pi + \varphi_0 \le \arg_\varphi(x) < -\pi + 2\varphi_0 \\ & \vdots \\ q_{A-1}^{(\varphi)} & -\pi + (A-1)\varphi_0 \le \arg_\varphi(x) < -\pi + A\varphi_0 \end{cases}, \quad (20)$$

$$\mathrm{csign}_B(x) = \begin{cases} q_0^{(\theta)} & -\frac{\pi}{2} \le \arg_\theta(x) < -\frac{\pi}{2} + \theta_0 \\ q_1^{(\theta)} & -\frac{\pi}{2} + \theta_0 \le \arg_\theta(x) < -\frac{\pi}{2} + 2\theta_0 \\ & \vdots \\ q_{B-1}^{(\theta)} & -\frac{\pi}{2} + (B-1)\theta_0 \le \arg_\theta(x) < -\frac{\pi}{2} + B\theta_0 \end{cases}, \quad (21)$$

$$\mathrm{csign}_C(x) = \begin{cases} q_0^{(\psi)} & -\frac{\pi}{4} \le \arg_\psi(x) < -\frac{\pi}{4} + \psi_0 \\ q_1^{(\psi)} & -\frac{\pi}{4} + \psi_0 \le \arg_\psi(x) < -\frac{\pi}{4} + 2\psi_0 \\ & \vdots \\ q_{C-1}^{(\psi)} & -\frac{\pi}{4} + (C-1)\psi_0 \le \arg_\psi(x) < -\frac{\pi}{4} + C\psi_0 \end{cases} \quad (22)$$

である．なお，$\arg_\varphi(x)$, $\arg_\theta(x)$, $\arg_\psi(x)$ はそれぞれ四元数 x を極座標表現した場合の位相 φ, θ, ψ を表している．

QBAAM は，可視層の初期状態 $\mathbf{x}(0)$ に入力パターンを設定し，不可視層と可視層のニューロンを交互に更新を繰り返すことでパターンの想起を行う．最終的に得られる可視層のニューロンの出力が QBAAM の出力パターンとなる．

3.2 学習

次に結合荷重の学習方法について説明する．QBAAM の学習に用いるパターンは，可視層の学習パターン (四元数) と不可視層の学習パターン (実数) の組み合わせから構成される．p 番目 $(p = 1, 2, \ldots, P)$ の可視層の学習パターン $\mathbf{x}^p$ を次のように定義する．

$$\mathbf{x}^p = (x_1^p, x_2^p, \ldots, x_N^p)^{\mathrm{T}}, \quad x_i^p \in \{q_{a,b,c}\}. \quad (23)$$

同様に，$\mathbf{x}^p$ に対応する不可視層の学習パターン $\mathbf{y}^p$ を

$$\mathbf{y}^p = (y_1^p, y_2^p, \ldots, y_M^p)^{\mathrm{T}}, \quad y_i^p \in \{1, -1\} \quad (24)$$

とする．なお，不可視層の学習パターンはランダムに与える．このとき，射影学習則による QBAAM の可視層から不可視層への結合荷重 $\mathbf{W}_{iv}$ と不可視層から可視層への結合荷重 $\mathbf{W}_{vi}$ は次式で定義される [12, 13]．

$$\mathbf{W}_{iv} = \mathbf{Y}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^*, \quad (25)$$

$$\mathbf{W}_{vi} = \mathbf{X}(\mathbf{Y}^{\mathrm{T}}\mathbf{Y})^{-1}\mathbf{Y}^{\mathrm{T}}. \quad (26)$$

ここで，$\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^P)$, $\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^P)$ である．また，$\mathbf{X}^*$ は $\mathbf{X}$ の随伴行列を表す．

4 数値実験

4.1 回転パターンの安定性

QBAAM と QHAM において回転パターンの安定性について調べた．本実験では，QBAAM の可視層の四元数ニューロン数 N を 2, 不可視層の実数ニューロン数 M を 2 とした．QHAM のニューロン数は QBAAM の可視層のニューロン数と同数に設定した．また，四元数ニューロン数の位相分割数は $(A, B, C) = (2, 2, 2)$ とした．学習パターンは $(q_{1,1,1}, q_{1,1,1})$ を用い，この学習パターンの各位相をシフトして得られる 7 種類の回転パターンを入力パターンとして想起を行った．なお，各層のニューロンの更新は可視層のニューロン出力が変化しなくなるまで繰り返した．

各入力パターンに対する QBAAM と QHAM の想起結果を Table 1 に示す．結果に示されるとおり，QBAAM では学習パターンとその反転パターン $(-q_{1,1,1}, -q_{1,1,1}) = (q_{0,1,1}, q_{0,1,1})$ のみが出力されており，反転パターン以外の回転パターンは想起されなかった．一方，QHAM では入力パターンとして用いた学習パターンとその回転パターンがそのまま出力されており，学習パターンとその回転パターンがすべて安定状態となった．以上の結果から，QBAAM の偽記憶は QHAM の偽記憶と比べて少ないことがわかる．

4.2 雑音耐性

次に QBAAM と QHAM の雑音耐性について調べた．本実験では，QBAAM の可視層のニューロン数 N を 100，不可視層のニューロン数 M を 100 とした．QHAM のニューロン数は QBAAM の可視層と同数に設定した．位相分割数は $(A, B, C) = (8, 4, 2)$, $(16, 8, 4)$, $(32, 16, 8)$，学習パターン数 P は 10, 30, 50 とした．学習パターンにはランダムに生成したパターンを用いた．入力パターンには学習パターンからランダムに選択したパターンに特定の割合で雑音を加えたパターンを用いた．

Figure 3 に，位相分割数と学習パターン数の各組み合わせにおける QBAAM と QHAM の雑音耐性を示す．各図の横軸は入力パターンに加える雑音の割合，縦軸は 1000 回の試行における想起成功率を示している．本実験では，出力パターンが入力パターンとして使用した元の学習パターンに完全に一致した場合を想起成功とした．同図に示されるとおり，QBAAM と QHAM はどちらも学習パターン数が増えると想起性能が低下しているが，どの条件においても QBAAM は QHAM より高い雑音耐性をもっている．また，位相分割数を増加させた場合，QBAAM の雑音耐性は維持されるが，QHAM の雑音耐性は大幅に低下している．QHAM の雑音耐性の低下は，位相分割数の増加にともなって偽記憶となる回転パターンが増加することに起因する．一方，QBAAM では偽記憶となる回転パターンは学習パターンの反転パターンに限られるため，その雑音耐性が位相分割数に依存しない．よって，QBAAM においては学習パターン数を増やした場合に限り雑音耐性が低下する．

Table 1: Stability of rotated patterns

Input	Output(QBAAM)	Output(QHAM)
$(q_{0,0,0}, q_{0,0,0})$	$(q_{0,1,1}, q_{0,1,1})$	$(q_{0,0,0}, q_{0,0,0})$
$(q_{0,0,1}, q_{0,0,1})$	$(q_{0,1,1}, q_{0,1,1})$	$(q_{0,0,1}, q_{0,0,1})$
$(q_{0,1,0}, q_{0,1,0})$	$(q_{0,1,1}, q_{0,1,1})$	$(q_{0,1,0}, q_{0,1,0})$
$(q_{0,1,1}, q_{0,1,1})$	$(q_{0,1,1}, q_{0,1,1})$	$(q_{0,1,1}, q_{0,1,1})$
$(q_{1,0,0}, q_{1,0,0})$	$(q_{1,1,1}, q_{1,1,1})$	$(q_{1,0,0}, q_{1,0,0})$
$(q_{1,0,1}, q_{1,0,1})$	$(q_{1,1,1}, q_{1,1,1})$	$(q_{1,0,1}, q_{1,0,1})$
$(q_{1,1,0}, q_{1,1,0})$	$(q_{1,1,1}, q_{1,1,1})$	$(q_{1,1,0}, q_{1,1,0})$
$(q_{1,1,1}, q_{1,1,1})$	$(q_{1,1,1}, q_{1,1,1})$	$(q_{1,1,1}, q_{1,1,1})$

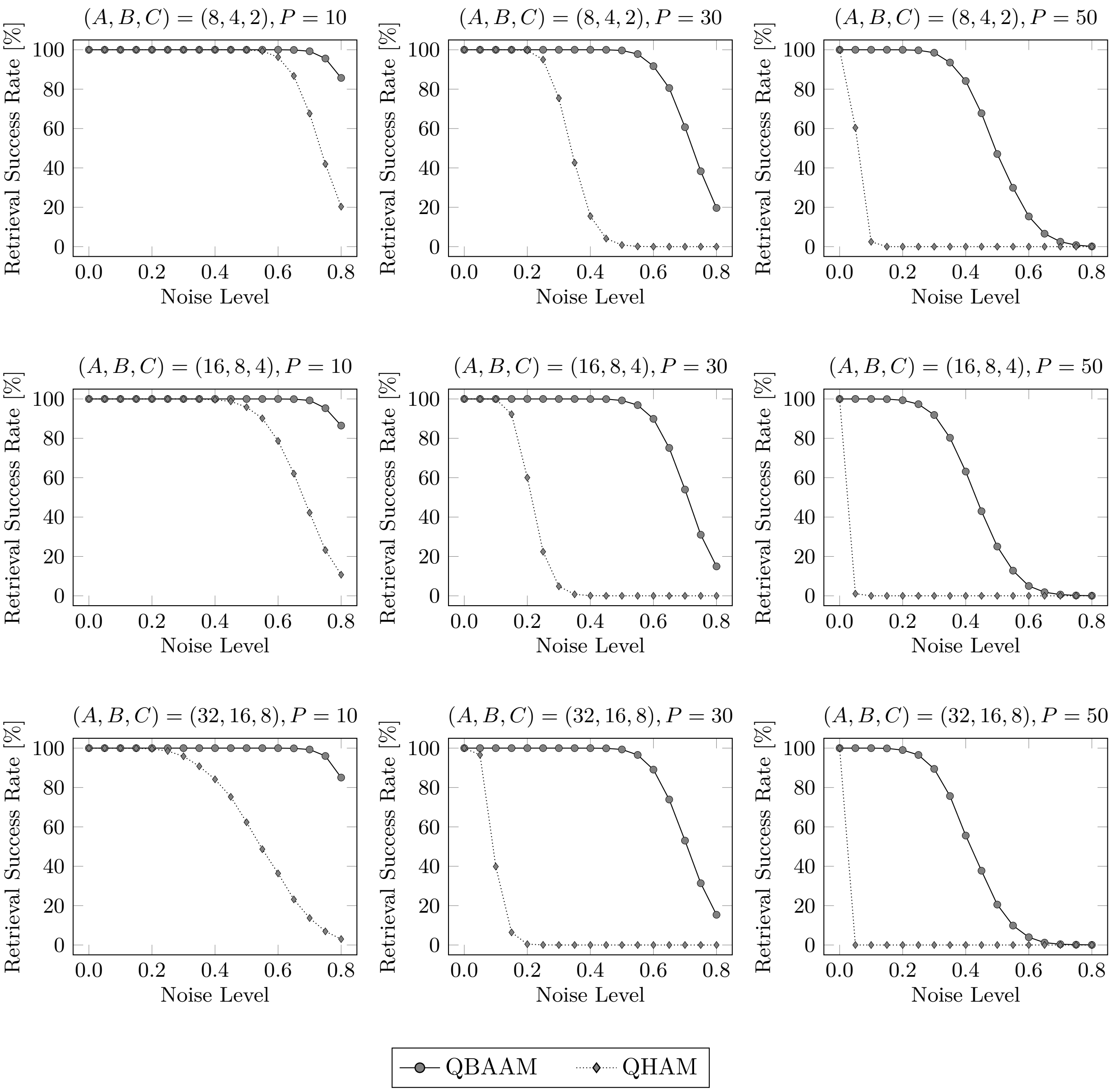


Fig. 3: Noise robustness of QBAAM and QHAM ($N = 100, M = 100$)

5 まとめ

本稿では，3 成分の多値パターンの想起を行うことができる QBAAM の想起性能を調査した．QBAAM は四元数ニューロンから構成される可視層と実数ニューロンから構成される不可視層の 2 層のネットワーク構造をもつ連想記憶モデルであり，不可視層の実数ニューロンの働きにより，学習パターンの回転パターン (偽記憶) が想起されることを防ぐ．数値実験では，単一の学習パターンを記銘した QBAAM において，回転パターンが想起されないことを確認した．また，ランダムパターンを用いた想起実験により，四元数ニューロンの位相分割数を増加させると QHAM の雑音耐性は低下するが，QBAAM の雑音耐性は維持されることを確認した．以上より，QBAAM は QHAM と比べて高い想起性能をもつことが示された．

今後の課題として，不可視層の学習パターンやニューロン数の雑音耐性に対する影響の調査やカラー画像のノイズ除去などへの応用が挙げられる．

参考文献

1) A. Hirose: Complex-valued neural networks: Theories and applications, World Scientific, 2003.
2) T. Nitta: Complex-valued neural networks: Utilizing high-dimensional parameters, IGI Global, 2009.
3) R. Mukundan: Quaternions: From Classical Mechanics to Computer Graphics, and Beyond, *Proceedings of the 7th Asian Technology Conference in Mathematics*, pp. 97–105, 2002.
4) P. Arena, L. Fortuna, G. Muscato, and M.G. Xibilia: Neural Networks in Multidimensional Domains, Vol. 234, Lecture Notes in Computer Science, Springer-Verlag, 1998.
5) H. Kusamichi, T. Isokawa, N. Matsui, Y. Ogawa, and K. Maeda: A New Scheme for Color Night Vision by Quaternion Neural Network, *Proceedings of the 2nd International Conference on Autonomous Robots and Agents (ICARA2004)*, pp. 101–106, 2004.
6) D.P. Mandic, C. Jahanchahi, and C.C. Took: A Quaternion Gradient Operator and Its Applications, *IEEE Signal Processing Letters*, Vol. 18, No. 1, pp. 47–50, 2011.
7) B.C. Ujang, C.C. Took, and D.P. Mandic: Quaternion-Valued Nonlinear Adaptive Filtering, *IEEE Transactions on Neural Networks*, Vol. 22, No. 8, pp. 1193–1206, 2011.
8) J.J. Hopfield: Neural networks and physical systems with emergent collective computational abilities, *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 79, No. 8, pp. 2554–2558, 1982.
9) T. Minemoto, T. Isokawa, H. Nishimura, and N. Matsui: Extended projection rule for quaternionic multistate hopfield neural network, *Proceedings of 20th International Symposium on Artificial Life and Robotics 2015 (AROB 20th 2015)*, pp. 418–423, 2015.
10) T. Bülow Hypercomplex spectral signal representations for the processing and analysis of images, PhD thesis, Christian Albrechts Universität, 1999.
11) T. Isokawa, H. Nishimura, and N. Matsui Quaternionic neural networks for associative memories, Complex-Valued Neural Networks:Advances and Applications, ed. by A. Hirose, pp. 103–131, Wiley-IEEE Press, 2013.
12) T. Minemoto, T. Isokawa, M. Kobayashi, H. Nishimura, and N. Matsui: On the performance of quaternionic bidirectional auto-associative memory, *Proceedings of 2015 International Join Conference on Neural Networks (IJCNN2015)*, pp. 2910–2915, 2015.
13) Y. Suzuki and M. Kobayashi: Complex-valued bipartite auto-associative memory, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. 97, No. 8, pp. 1680–1687, 2014.
14) N.N. Aizenberg, Y.L. Ivaskiv, and D.A. Pospelov: About one generalization of the threshold function, *Doklady Akademii Nauk SSSR (The Reports of the Academy of Sciences of the USSR)*, Vol. 196, No. 6, pp. 1287–1290, 1971. (in Russian).

双曲勾配オペレータと双曲誤差逆伝播アルゴリズム

○新田 徹（産業技術総合研究所）　黒江 康明（京都工芸繊維大学）

Hyperbolic Gradient Operator and Hyperbolic Backpropagation Algorithm

*T. Nitta (National Institute of Advanced Industrial Science and Technology, AIST)
and Y. Kuroe (Kyoto Institute of Technology)

Abstract - In this paper, we define the Wirtinger derivative for hyperbolic functions, and derive the hyperbolic gradient operator using it. Furthermore, we derive the hyperbolic back-propagation algorithms for some multi-layered hyperbolic neural networks using the hyperbolic gradient operator. The use of the Wirtinger derivative reduces the efforts required the derivation of the learning algorithms to half, simplifies the representation of the learning algorithms, and makes their computer programs easier to code.

Key Words: Wirtinger Calculus, Hyperbolic number, Neural network, Learning algorithm, Backpropagation

1　はじめに

機械学習の手法の中には，非正則な双曲関数の微分計算が必要となるものが存在する．たとえば，階層型双曲ニューラルネットワーク（以下，ニューラルネットワークを NN と略する）における誤差関数は実数値関数であるため，非正則であり，双曲微分可能ではない．そのため，誤差関数をパラメータの実部と Unipotent 部に関してそれぞれ微分する必要がある．これまで，非正則な複素関数および四元数関数に対する微分計算を容易にする Wirtinger Calculus [1, 2, 3, 4, 5, 6, 7] は提供されている．そして，複素 NN の学習アルゴリズムなどを導出するのに使われている．たとえば，Amin らは，一般的な活性化関数 f を持つ複素 NN に対する複素 BP 則を Wirtinger 微分を使って導出している [8]．

双曲関数に対する Wirtinger 微分はまだ提案されていない．そこで本研究では，双曲関数に対する Wirtinger 微分を定義し，それを用いて双曲勾配オペレータを導出する．また，導いた双曲勾配オペレータを使って，いくつかの３層双曲 NN モデルを対象にして，双曲誤差逆伝播学習アルゴリズムを導く．Wirtinger 微分を用いると，学習アルゴリズムを導出するのにかかる計算の労力は半分で済む，学習則の表現が単純になる，プログラムのコーディングが容易になるといった利点がある．

2　Wirtinger 微分

本章では，文献 6) に沿って Wirtinger 微分について述べる．

複素平面 $\boldsymbol{C}$ 上の領域 D で定義された複素関数 $f : D \to \boldsymbol{C}, z = x + iy \mapsto f(z)$ を考える．$\boldsymbol{C}$ は複素数全体の集合を表す．この複素関数の実部 $Re[f]$ と虚部 $Im[f]$ は，それぞれ実２変数 (x, y) の関数とみなすことができる．そして，$Re[f]$ および $Im[f]$ が点 (x_0, y_0) で全微分可能であるためには，次の条件が必要かつ十分である．

条件 1 複素関数 $f_1 : D \to \boldsymbol{C}$，$f_2 : D \to \boldsymbol{C}$ で，点 $z_0 \in D$ で連続なものが存在して，各点 $z \in D$ で，

$$f(z) = f(z_0) + (z - z_0)f_1(z) + (z^* - z_0^*)f_2(z) \tag{1}$$

が成り立ち，$f_1(z_0)$ および $f_2(z_0)$ は一意に決まる．□

以下，本章で考える複素関数 $f : D \to \boldsymbol{C}, z = x + iy \mapsto f(z)$ は条件 1 を満たすものとする．

このとき，Wirtinger 微分は次のように定義される．

定義 1 複素関数 f の点 $z_0 \in D$ における Wirtinger 微分係数 $\partial f/\partial z(z_0)$, $\partial f/\partial z^*(z_0)$ を次のように定義する．

$$\frac{\partial f}{\partial z}(z_0) = f_1(z_0), \tag{2}$$

$$\frac{\partial f}{\partial z^*}(z_0) = f_2(z_0). \tag{3}$$

ここで，f_1 および f_2 は条件 1 において定まるものである． □

このとき，次の定理が成り立つ．

定理 1

$$\frac{\partial f}{\partial z}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)-i\frac{\partial f}{\partial y}(z_0)\right), \tag{4}$$

$$\frac{\partial f}{\partial z^*}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)+i\frac{\partial f}{\partial y}(z_0)\right). \tag{5}$$

□

Wirtinger 微分を用いると，複素関数の正則性を次のように述べることができる．

定理 2 複素関数 $f:D\to \boldsymbol{C}$ が正則であるためには，領域 D の各点 z_0 において

$$\frac{\partial f}{\partial z^*}(z_0)=0 \tag{6}$$

が成り立つことが必要十分である． □

(注意) Brandwood の論文 [2] では，Witringer 微分に関して次のように述べられているが，誤解を与える可能性があるので，注意する必要がある．

[Brandwood 論文 [2] における定理 1]

複素関数 $f:\boldsymbol{C}\to\boldsymbol{C}, z=x+iy\mapsto f(z)$ は，$f:\boldsymbol{R}\times\boldsymbol{R}\to\boldsymbol{C},(x,y)\mapsto f(x,y)$ とみなすことができる（$\boldsymbol{R}$ は実数全体の集合）．$g:\boldsymbol{C}\times\boldsymbol{C}\to\boldsymbol{C},(z,z^*)\mapsto g(z,z^*)$ が存在して，$g(z,z^*)$ は z および z^* に関して独立に（他方を定数とみなして）それぞれ analytic であり，$g(z,z^*)=f(x,y)$ が成り立つと仮定する．このとき，

$$\frac{\partial f}{\partial z}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)-i\frac{\partial f}{\partial y}(z_0)\right), \tag{7}$$

$$\frac{\partial f}{\partial z^*}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)+i\frac{\partial f}{\partial y}(z_0)\right). \tag{8}$$

□

上記の定理では，*analytic* という用語が使われているが，これを「正則」と解釈した場合，条件が厳しすぎる．よって，条件 1 が成り立つこと，つまり「全微分可能」と解釈するのが妥当と思われる．実際，そのように解釈しても上記定理は成り立つ．

3 双曲勾配オペレータの導出

本章では，第 2 章で述べた Wirtinger 微分を双曲関数に拡張したうえで，最急降下方向を表す双曲勾配オペレータを導出する．

3.1 双曲数

双曲数は次の形をした数である．

$$z=a+ub, \tag{9}$$

ここで，$a,b\in\boldsymbol{R}$ であり，u は $u\neq\pm1$，$u^2=1$ を満たし，*unipotent* と呼ばれる [9]．双曲数の乗法は $(a+ub)(c+ud)\overset{\mathrm{def}}{=}(ac+bd)+u(ad+bc)$ と定義される．双曲数 $z=a+ub\in\boldsymbol{H}$ に対して，その双曲共役数は $z^*\overset{\mathrm{def}}{=}a-ub$，絶対値は $|z|_h\overset{\mathrm{def}}{=}\sqrt{|a^2-b^2|}$ とそれぞれ定義される（$\boldsymbol{H}$ は双曲数全体の集合）．もし $|a|=|b|$ ならば z の絶対値は 0 になる．

双曲数は 3 次方程式の解の表現や特殊相対論における座標軸に使われている [9]．

3.2 Wirtinger 微分の双曲関数への拡張

第 2 章で述べた Wirtinger 微分を双曲関数に拡張する．

双曲平面 $\boldsymbol{H}$ 上の領域 D で定義された双曲関数 $f:D\to\boldsymbol{H}, z=x+uy\mapsto f(z)$ を考える．この双曲関数の実部 $Re[f]$ と Unipotent 部 $Un[f]$ は，それぞれ実 2 変数 (x,y) の関数とみなすことができる．そして，$Re[f]$ および $Un[f]$ が点 (x_0,y_0) で全微分可能であるためには，次の条件が必要かつ十分である．

条件 2 双曲関数 $f_1:D\to\boldsymbol{H}$，$f_2:D\to\boldsymbol{H}$ で，点 $z_0\in D$ で連続なものが存在して，各点 $z\in D$ で，

$$f(z)=f(z_0)+(z-z_0)f_1(z)+(z^*-z_0^*)f_2(z) \tag{10}$$

が成り立ち，$f_1(z_0)$ および $f_2(z_0)$ は一意に決まる． □

以下，本章で考える双曲関数 $f:D\to\boldsymbol{H},z=x+uy\mapsto f(z)$ は条件 2 を満たすものとする．つまり，実部 $Re[f(x,y)]$ と Unipotent 部 $Un[f(x,y)]$ をそれぞれ実 2 変数 (x,y) の実数値関数とみた時，$Re[f(x,y)]$ および $Un[f(x,y)]$ が点 (x_0,y_0) で全微分可能とする．

双曲 Wirtinger 微分を次のように定義する．

定義 2 双曲関数 $f:D\to\boldsymbol{H},z=x+uy\mapsto f(z)$ の点 $z_0\in D$ における双曲 Wirtinger 微分係数 $\partial f/\partial z(z_0), \partial f/\partial z^*(z_0)$ を次のように定義する．

$$\frac{\partial f}{\partial z}(z_0)=f_1(z_0), \tag{11}$$

$$\frac{\partial f}{\partial z^*}(z_0)=f_2(z_0). \tag{12}$$

ここで，f_1 および f_2 は条件 2 において定まるものである． □

双曲 Wirtinger 微分は次のように計算できる．

定理 3 次の式が成り立つ．

$$\frac{\partial f}{\partial z}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)+u\frac{\partial f}{\partial y}(z_0)\right), \quad (13)$$

$$\frac{\partial f}{\partial z^*}(z_0)=\frac{1}{2}\left(\frac{\partial f}{\partial x}(z_0)-u\frac{\partial f}{\partial y}(z_0)\right). \quad (14)$$

□

(証明) $Re[f(x,y)], Un[f(x,y)]$ の全微分可能性から容易に導かれる．■

双曲 Wirtinger 微分を用いると，双曲関数の正則性を次のように述べることができる．

定理 4 双曲関数 $f: D \to \boldsymbol{H}$ が正則であるためには，領域 D の各点 z_0 において

$$\frac{\partial f}{\partial z^*}(z_0)=0 \quad (15)$$

が成り立つことが必要十分である． □

(証明) 紙面の制約により省略．■

3.3 最急降下方向の双曲勾配オペレータ

任意の $\boldsymbol{z}=(z_1,\cdots,z_N)^T \in \boldsymbol{H}^N$ に対して，

$$\nabla_{\boldsymbol{z}} \overset{\text{def}}{=} \left(\frac{\partial}{\partial z_1}\cdots\frac{\partial}{\partial z_N}\right)^T, \quad (16)$$

$$\nabla_{\boldsymbol{z}^*} \overset{\text{def}}{=} \left(\frac{\partial}{\partial z_1^*}\cdots\frac{\partial}{\partial z_N^*}\right)^T \quad (17)$$

とし，$\nabla_{\boldsymbol{z}}$ を $\boldsymbol{z}$ に関する双曲勾配オペレータ，$\nabla_{\boldsymbol{z}^*}$ を双曲共役勾配オペレータとそれぞれ呼ぶ．ここで，N は自然数であり，T は転置を表す．

双曲勾配オペレータを使うと，実数値双曲関数 $f: D^N \to \boldsymbol{R}, \boldsymbol{z}=(z_1,\cdots,z_N)^T \mapsto f(\boldsymbol{z})$ が停留状態 (各変数 z_k の実部と Unipotent 部の微小変化に対する f の変化量が 0 である状態) にあるための必要十分条件を次のように表現することができる：$\nabla_{\boldsymbol{z}} f=\boldsymbol{0}$ または $\nabla_{\boldsymbol{z}^*} f=\boldsymbol{0}$.

次の定理は，実数値双曲関数 $f: D^N \to \boldsymbol{R}, \boldsymbol{z}=(z_1,\cdots,z_N)^T \mapsto f(\boldsymbol{z})$ の最急降下方向が双曲勾配オペレータによって与えられることを示す．

定理 5 実数値双曲関数 $f: D^N \to \boldsymbol{R}, \boldsymbol{z}=(z_1,\cdots,z_N)^T \mapsto f(\boldsymbol{z})$ は各 $z_k \in D$ に関して，条件 2 を満たしているとする $(1 \le k \le N)$．このとき，(a) $\boldsymbol{z}=\nabla_{\boldsymbol{z}} f$ は $f(\boldsymbol{z})$ が最も大きく変化する方向である．(b) $\boldsymbol{z}$ が $\delta\boldsymbol{z}$ だけ変化するとき，f の変化量 δf は次の式で与えられる．

$$\delta f = 2Re\Big[(\nabla_{\boldsymbol{z}^*} f)^H \delta\boldsymbol{z}\Big], \quad (18)$$

ここで，H は双曲共役転置を表す． □

(証明)

$$\begin{aligned}
\delta f &= \sum_{k=1}^{N}\left(\frac{\partial f}{\partial z_k}\delta z_k+\frac{\partial f}{\partial z_k^*}\delta z_k^*\right)\\
&= (\nabla_{\boldsymbol{z}} f)^T\delta\boldsymbol{z}+(\nabla_{\boldsymbol{z}^*} f)^T\delta\boldsymbol{z}^*\\
&= (\nabla_{\boldsymbol{z}} f)^T\delta\boldsymbol{z}+[(\nabla_{\boldsymbol{z}} f)^T\delta\boldsymbol{z}]^*\\
&\qquad (\nabla_{\boldsymbol{z}^*} f=(\nabla_{\boldsymbol{z}} f)^* \text{による})\\
&= 2Re\Big[(\nabla_{\boldsymbol{z}} f)^T\delta\boldsymbol{z}\Big]\\
&= 2Re\Big[(\nabla_{\boldsymbol{z}^*} f)^H\delta\boldsymbol{z}\Big]\\
&\qquad (\nabla_{\boldsymbol{z}^*} f=(\nabla_{\boldsymbol{z}} f)^* \text{による})\\
&\le 2\cdot \max_{\substack{\delta\boldsymbol{z}\in\boldsymbol{H}^N\\ \|\delta\boldsymbol{z}\|\ \text{は一定}}} Re\Big[(\nabla_{\boldsymbol{z}^*} f)^H\delta\boldsymbol{z}\Big]\\
&= 2\|\nabla_{\boldsymbol{z}^*} f\|\cdot\|\delta\boldsymbol{z}\|, \qquad (19)
\end{aligned}$$

ここで，$\|w\| \overset{\text{def}}{=} \sqrt{a^2+b^2}, w=a+ub$．最後の等式を与えるのは，$\delta\boldsymbol{z}=k(\nabla_{\boldsymbol{z}} f)$ を満たす $\delta\boldsymbol{z} \in \boldsymbol{H}^N$ である（k はある正の実数）．これで (a) が言えた．■

複素関数および四元数関数に対する最急降下法に関する定理の証明には，シュバルツの不等式が使われている [2, 7]．しかし，双曲関数の場合は内積が定義できないため，シュバルツの不等式は使わない証明になっている．

次の定理は，双曲共役勾配オペレータ (式 (17)) の方向を明らかにする．

定理 6 実数値双曲関数 $f: D^N \to \boldsymbol{R}, \boldsymbol{z}=(z_1,\cdots,z_N)^T \mapsto f(\boldsymbol{z})$ は各 $z_k \in D$ に関して，条件 2 を満たしているとする $(1 \le k \le N)$．ここで，N は自然数である．このとき，$\nabla_{\boldsymbol{z}^*} f$ は，次の式が成り立つ時，$f(\boldsymbol{z})$ の上昇方向であり，成り立たない時，下降方向である：

$$\sum_{k=1}^{N}\left(\frac{\partial f(\boldsymbol{x},\boldsymbol{y})}{\partial x_k}\right)^2 < \sum_{k=1}^{N}\left(\frac{\partial f(\boldsymbol{x},\boldsymbol{y})}{\partial y_k}\right)^2. \quad (20)$$

また，$\boldsymbol{z}=\nabla_{\boldsymbol{z}^*}f$ が最急降下方向であるための必要十分条件は

$$\frac{\partial f(\boldsymbol{x},\boldsymbol{y})}{\partial y_k}=0 \quad (1\le k\le N) \tag{21}$$

である． □

(証明) 紙面の制約により省略． ■

4 双曲ニューラルネットワークへの応用

本章では，3 章で求めた双曲勾配オペレータを用いて，多層双曲ニューラルネットワークの誤差逆伝播学習アルゴリズム（双曲 BP 則）を求める．

4.1 多層双曲ニューラルネットワークモデル

本節では，解析の対象とする多層双曲 NN について述べる．

まず，次のような双曲ニューロン n を考える．入力信号，重み，閾値，出力信号はすべて双曲数であり，入力の総和は $U_n=\sum_m W_{nm}X_m+V_n$, で与えられる．ここで，W_{nm} は双曲ニューロン n と m の間の（双曲数値）重みであり，X_m は双曲ニューロン m からの（双曲数値）入力信号，V_n は双曲ニューロン n の（双曲数値）閾値である．（双曲数値）出力信号を得るために，入力の総和 U_n に双曲活性化関数 $f:\boldsymbol{H}\to\boldsymbol{H}$ を適用する．（双曲数値）出力信号は $f(U_n)$ で与えられる．

双曲 NN は上記のような双曲ニューロンから構成される．対象とする双曲 NN は簡単のため，3 層 $L-M-N$ ネットワークとする．入力パターン $\boldsymbol{I}=(I_1,\cdots,I_L)^T\in\boldsymbol{H}^L$ に対して，出力ニューロン k の出力値 O_k は

$$O_k = f(S_k), \tag{22}$$

$$S_k = \sum_{j=1}^{M} v_{kj}H_j+\gamma_k, \tag{23}$$

$$H_j = f(U_j), \tag{24}$$

$$U_j = \sum_{i=1}^{L} w_{ji}I_i+\theta_j \tag{25}$$

で与えられる．ここで，$I_i\in\boldsymbol{H}$ は入力ニューロン i への入力信号，$w_{ji}\in\boldsymbol{H}$ は入力ニューロン i と中間ニューロン j の間の重み，$\theta_j\in\boldsymbol{H}$ は中間ニューロン j の閾値，$v_{kj}\in\boldsymbol{H}$ は中間ニューロン j と出力ニューロン k の間の重み，$\gamma_k\in\boldsymbol{H}$ は出力ニューロン k の閾値である $(1\le i\le L, 1\le j\le M, 1\le k\le N)$．また，$\boldsymbol{z}=(z_1,\cdots,z_K)^T\in\boldsymbol{H}^K$ をすべての学習パラメータを一つのベクトルにまとめたものとする．ここで，$z_n\in\boldsymbol{H}$ は学習パラメータ $\{w_{ji},\theta_j,v_{jk},\gamma_k\}$ のいずれかを表す $(1\le n\le K)$．

P 組の双曲数値学習データ $\{(\boldsymbol{I}^{(p)},\boldsymbol{T}^{(p)})\in\boldsymbol{H}^L\times\boldsymbol{H}^N \mid p=1,\cdots,P\}$ が与えられたとき，それらによって表現される関係を実現するのに双曲 NN を使う．学習の目的は，

$$E(\boldsymbol{z})=\sum_{p=1}^{P}\|T_k-O_k\|^2\in\boldsymbol{R} \tag{26}$$

によって定義される誤差関数を最小化する学習パラメータを見つけることである．式 (26) の誤差関数は実数値関数なので，双曲関数として正則ではない．式 (26) の誤差関数は次のように書き直せる．

$$E(\boldsymbol{z},\boldsymbol{z}^*)=\frac{1}{2}\sum_{p=1}^{P}\left[(T_k-O_k)^2+(T_k^*-O_k^*)^2\right]. \tag{27}$$

4.2 双曲誤差逆伝播アルゴリズム

本節では，様々な双曲活性化関数を持つ３層双曲ＮＮの双曲ＢＰ則を導出する．

4.2.1 一般的な活性化関数の場合

まず，一般的な活性化関数を持つモデルの双曲ＢＰ則を求める．任意のパラメータ α に対して，その更新は次の式によって行われる：

$$\alpha_{n+1}=\alpha_n+\Delta\alpha, \tag{28}$$

ここで，α_n は n 回目の更新を行った後のパラメータ α の値，$\Delta\alpha$ は更新量をそれぞれ表す．

定理 5 により，誤差関数 E が最も大きく変化する方向は，

$$\begin{aligned}\boldsymbol{z} &= \nabla_{\boldsymbol{z}}E(\boldsymbol{z},\boldsymbol{z}^*)\\ &= \left(\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_1},\cdots,\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_K}\right)^T\end{aligned} \tag{29}$$

である．

$\varepsilon>0$ を学習率として，入力ニューロン i と中間ニューロン j の間の重み w_{ji} の更新量 Δw_{ji}

は，式 (29) に従えば次のように計算できる．

$$\begin{aligned}\Delta w_{ji} &= -\varepsilon\cdot\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial w_{ji}}\\ &= \varepsilon\sum_{k=1}^{N}\left[(T_k-O_k)\frac{\partial O_k}{\partial w_{ji}}\right.\\ &\quad\left.+(T_k-O_k)^*\cdot\frac{\partial O_k^*}{\partial w_{ji}}\right]. \end{aligned}\qquad(30)$$

ここで，

$$\begin{aligned}\frac{\partial O_k}{\partial w_{ji}} &= \frac{\partial f(S_k,S_k^*)}{\partial w_{ji}}\\ &= \frac{\partial f(S_k,S_k^*)}{\partial S_k}\cdot\frac{\partial S_k}{\partial w_{ji}}\\ &\quad+\frac{\partial f(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial S_k^*}{\partial w_{ji}},\end{aligned}\qquad(31)$$

$$\begin{aligned}\frac{\partial O_k^*}{\partial w_{ji}} &= \frac{\partial f^*(S_k,S_k^*)}{\partial w_{ji}}\\ &= \frac{\partial f^*(S_k,S_k^*)}{\partial S_k}\cdot\frac{\partial S_k}{\partial w_{ji}}\\ &\quad+\frac{\partial f^*(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial S_k^*}{\partial w_{ji}}.\end{aligned}\qquad(32)$$

さらに，

$$\begin{aligned}\frac{\partial S_k}{\partial w_{ji}} &= v_{kj}\frac{\partial f(U_j,U_j^*)}{\partial w_{ji}}\\ &= v_{kj}I_i\cdot\frac{\partial f(U_j,U_j^*)}{\partial U_j},\end{aligned}\qquad(33)$$

$$\begin{aligned}\frac{\partial S_k^*}{\partial w_{ji}} &= v_{kj}^*\frac{\partial f^*(U_j,U_j^*)}{\partial w_{ji}}\\ &= v_{kj}^*I_i\cdot\frac{\partial f^*(U_j,U_j^*)}{\partial U_j}.\end{aligned}\qquad(34)$$

よって，式 (30) - (34) から，

$$\begin{aligned}\Delta w_{ji} &= \varepsilon I_i\sum_{k=1}^{N}\left[\delta_k\left\{v_{kj}\frac{\partial f(S_k,S_k^*)}{\partial S_k}\right.\right.\\ &\quad\cdot\frac{\partial f(U_j,U_j^*)}{\partial U_j}\\ &\quad\left.+v_{kj}^*\frac{\partial f(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial f^*(U_j,U_j^*)}{\partial U_j}\right\}\\ &\quad+\delta_k^*\left\{v_{kj}\frac{\partial f^*(S_k,S_k^*)}{\partial S_k}\cdot\frac{\partial f(U_j,U_j^*)}{\partial U_j}\right.\\ &\quad\left.\left.+v_{kj}^*\frac{\partial f^*(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial f^*(U_j,U_j^*)}{\partial U_j}\right\}\right],\end{aligned}\qquad(35)$$

ここで，$\delta_k \overset{\mathrm{def}}{=} T_k - O_k$ とおいた．

同様にして，中間ニューロン j と出力ニューロン k の間の重み v_{kj} の更新量 Δv_{kj} は，式 (29) に従えば次のように計算できる．

$$\begin{aligned}\Delta v_{kj} &= -\varepsilon\cdot\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial v_{kj}}\\ &= \varepsilon\left[\delta_k\frac{\partial O_k}{\partial v_{kj}}+\delta_k^*\cdot\frac{\partial O_k^*}{\partial v_{kj}}\right]\\ &= \varepsilon H_j\left[\delta_k\frac{\partial f(S_k,S_k^*)}{\partial S_k}\right.\\ &\quad\left.+\delta_k^*\frac{\partial f^*(S_k,S_k^*)}{\partial S_k}\right].\end{aligned}\qquad(36)$$

閾値は入力値が 1 である場合の重みとみなせるから省略する．

4.2.2 シグモイド活性化関数の場合

シグモイド関数

$$f(z)=\frac{1}{1+e^{-z}},\qquad z\in\boldsymbol{H}\qquad(37)$$

を活性化関数として持つモデルの双曲ＢＰ則を求める．この関数は双曲関数として正則であり，かつ，有界である [10)]．

入力ニューロン i と中間ニューロン j の間の重み w_{ji} の更新量 Δw_{ji} は，式 (35) から求められる．シグモイド活性化関数の場合，

$$\frac{\partial f(S_k,S_k^*)}{\partial S_k} = O_k(1-O_k),\qquad(38)$$

$$\frac{\partial f(S_k,S_k^*)}{\partial S_k^*} = \frac{\partial f^*(S_k,S_k^*)}{\partial S_k}=0,\qquad(39)$$

$$\frac{\partial f(U_j,U_j^*)}{\partial U_j} = H_j(1-H_j),\qquad(40)$$

$$\frac{\partial f^*(U_j,U_j^*)}{\partial U_j} = 0\qquad(41)$$

であるから，

$$\Delta w_{ji} = \varepsilon I_iH_j(1-H_j)\sum_{k=1}^{N}\delta_kv_{kj}O_k(1-O_k)\qquad(42)$$

となる．

また，中間ニューロン j と出力ニューロン k の間の重み v_{kj} の更新量 Δv_{kj} は，式 (36) から求められる．式 (38)，(39) より，

$$\Delta v_{kj} = \varepsilon\delta_kH_jO_k(1-O_k)\qquad(43)$$

となる．

4.2.3 分離型シグモイド活性化関数の場合

分離型シグモイド関数

$$f(z) = \frac{1}{1+e^{-x}} + u\frac{1}{1+e^{-y}}, \quad z = x + uy \in \boldsymbol{H} \tag{44}$$

を活性化関数として持つモデル[11]の双曲ＢＰ則を求める．この関数は有界である．そして，$\{z \in \boldsymbol{H} |\ |z| = 0\}$ 上で双曲関数として正則であるが，それ以外では正則でない．

入力ニューロン i と中間ニューロン j の間の重み w_{ji} の更新量 Δw_{ji} は，式 (35) から求められる．分離型シグモイド活性化関数の場合，

$$\frac{\partial f(S_k,S_k^*)}{\partial S_k} = \frac{1}{2}\left[O_k^R(1-O_k^R) + O_k^U(1-O_k^U)\right], \tag{45}$$

$$\frac{\partial f^*(S_k,S_k^*)}{\partial S_k} = \frac{1}{2}\left[O_k^R(1-O_k^R) - O_k^U(1-O_k^U)\right], \tag{46}$$

$$\frac{\partial f(U_j,U_j^*)}{\partial U_j} = \frac{1}{2}\left[H_j^R(1-H_j^R) + H_j^U(1-H_j^U)\right], \tag{47}$$

$$\frac{\partial f^*(U_j,U_j^*)}{\partial U_j} = \frac{1}{2}\left[H_j^R(1-H_j^R) - H_j^U(1-H_j^U)\right] \tag{48}$$

であるから，

$$\begin{aligned}\Delta w_{ji} = \frac{\varepsilon}{2} I_i \Bigg[& \left\{H_j^R(1-H_j^R) + H_j^U(1-H_j^U)\right\} \cdot \\ & \sum_{k=1}^{N} v_{kj}\left\{\delta_k^R O_k^R(1-O_k^R) + u\cdot\delta_k^U O_k^U(1-O_k^U)\right\} \\ & + \left\{H_j^R(1-H_j^R) - H_j^U(1-H_j^U)\right\} \cdot \\ & \sum_{k=1}^{N} v_{kj}^*\left\{\delta_k^R O_k^R(1-O_k^R) \right. \\ & \left. - u\cdot\delta_k^U O_k^U(1-O_k^U)\right\}\Bigg]\end{aligned} \tag{49}$$

となる．

また，中間ニューロン j と出力ニューロン k の間の重み v_{kj} の更新量 Δv_{kj} は，式 (36) から求められる．式 (45)，(46) より，

$$\begin{aligned}\Delta v_{kj} = \varepsilon H_j \Big\{ & \delta_k^R O_k^R(1-O_k^R) \\ + \ & u\cdot\delta_k^U O_k^U(1-O_k^U)\Big\}\end{aligned} \tag{50}$$

となる．

4.2.4 Hitzer 活性化関数の場合

Hitzer が提案した活性化関数[12)]

$$f(z) = \frac{1}{1+e^{-x-y}} + u\frac{1}{1+e^{-x-y}}, \quad z = x + uy \in \boldsymbol{H} \tag{51}$$

を持つモデルの双曲ＢＰ則を求める．この関数は双曲関数として正則であり，かつ，有界である．

入力ニューロン i と中間ニューロン j の間の重み w_{ji} の更新量 Δw_{ji} は，式 (35) から求められる．Hirzer 活性化関数の場合，

$$\frac{\partial f(S_k,S_k^*)}{\partial S_k} = (1+u)O_k^R(1-O_k^R), \tag{52}$$

$$\frac{\partial f(S_k,S_k^*)}{\partial S_k^*} = \frac{\partial f^*(S_k,S_k^*)}{\partial S_k} = 0, \tag{53}$$

$$\frac{\partial f(U_j,U_j^*)}{\partial U_j} = (1+u)H_j^R(1-H_j^R), \tag{54}$$

$$\frac{\partial f^*(U_j,U_j^*)}{\partial U_j} = 0 \tag{55}$$

であるから，

$$\Delta w_{ji} = 2\varepsilon(1+u)I_i H_j^R(1-H_j^R) \sum_{k=1}^{N} \delta_k v_{kj} O_k^R(1-O_k^R) \tag{56}$$

となる．

また，中間ニューロン j と出力ニューロン k の間の重み v_{kj} の更新量 Δv_{kj} は，式 (36) から求められる．式 (52)，(53) より，

$$\Delta v_{kj} = \varepsilon H_j \delta_k (1+u) O_k^R(1-O_k^R) \tag{57}$$

となる．

4.3 考察

4.3.1 双曲 Wirtinger 微分の利点

4.2 節で導いた学習アルゴリズムは，通常，誤差関数をパラメータの実部と Unipotent 部に関してそれぞれ偏微分して求める．つまり，任意のパラメータ $z_n \in \boldsymbol{H}$ の更新則を

$$\Delta z_n = -\varepsilon\cdot\left(\frac{\partial E(\boldsymbol{z})}{\partial z_n^R} + u\frac{\partial E(\boldsymbol{z})}{\partial z_n^U}\right) \tag{58}$$

を計算することによって求める．ここで，z_n^R，z_n^U はそれぞれパラメータ z_n の実部，Unipotent 部である．一方，双曲 Wirtinger 微分の場合，双曲勾配オペレータを使って，

$$\Delta z_n = -\varepsilon \cdot \frac{\partial E(\boldsymbol{z})}{\partial z_n} \tag{59}$$

を計算することによって求められるので，計算の労力は前者に比べて半分で済む．また，学習則の表現が単純になり，プログラムのコーディングが容易になる（詳細は紙面の都合により省略する）．

4.3.2 複素 BP 則との比較

次に，4.2.1 節で求めた双曲 BP 則を複素 BP 則と比較する．

比較の対象とするのは，３層複素 $L-M-N$ ネットワークである．入力信号，重み，閾値，出力信号はすべて複素数であり，活性化関数は $f:\boldsymbol{C}\to\boldsymbol{C}$ である．入力パターン $\boldsymbol{I}=(I_1,\cdots,I_L)^T\in\boldsymbol{C}^L$ に対して，出力ニューロン k の出力値 O_k は

$$\begin{aligned} O_k &= f(S_k), && (60)\\ S_k &= \sum_{j=1}^{M} v_{kj}H_j+\gamma_k, && (61)\\ H_j &= f(U_j), && (62)\\ U_j &= \sum_{i=1}^{L} w_{ji}I_i+\theta_j && (63) \end{aligned}$$

で与えられる．ここで，$I_i\in\boldsymbol{C}$ は入力ニューロン i への入力信号，$w_{ji}\in\boldsymbol{C}$ は入力ニューロン i と中間ニューロン j の間の重み, $\theta_j\in\boldsymbol{C}$ は中間ニューロン j の閾値，$v_{kj}\in\boldsymbol{C}$ は中間ニューロン j と出力ニューロン k の間の重み, $\gamma_k\in\boldsymbol{C}$ は出力ニューロン k の閾値である $(1\le i\le L, 1\le j\le M, 1\le k\le N)$．また，$\boldsymbol{z}=(z_1,\cdots,z_K)^T\in\boldsymbol{C}^K$ をすべての学習パラメータを一つのベクトルにまとめたものとする．ここで，$z_n\in\boldsymbol{C}$ は学習パラメータ $\{w_{ji},\theta_j,v_{jk},\gamma_k\}$ のいずれかを表す $(1\le n\le K)$．誤差関数を

$$E(\boldsymbol{z})=\sum_{p=1}^{P}\|T_k-O_k\|^2\in\boldsymbol{R} \tag{64}$$

とする．ここで，P は学習パターンの個数，$T_k\in\boldsymbol{C}$ は出力ニューロン k に対する教師信号である．

式 (64) は，次のように書き直せる．

$$E(\boldsymbol{z},\boldsymbol{z}^*)=\sum_{p=1}^{P}(T_k-O_k)(T_k^*-O_k^*). \tag{65}$$

文献 2) により，誤差関数の最急降下方向は

$$\begin{aligned} \boldsymbol{z} &= \nabla_{\boldsymbol{z}^*}E(\boldsymbol{z},\boldsymbol{z}^*)\\ &= \left(\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_1^*},\cdots,\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_K^*}\right)^T \end{aligned} \tag{66}$$

である．式 (66) に従って計算することにより，次の学習則を得る．

$$\begin{aligned} \Delta w_{ji} &= \varepsilon I_i^*\sum_{k=1}^{N}\Bigg[\delta_k\Bigg\{v_{kj}^*\frac{\partial f^*(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial f^*(U_j,U_j^*)}{\partial U_j^*}\\ &\quad +v_{kj}\frac{\partial f^*(S_k,S_k)}{\partial S_k}\cdot\frac{\partial f(U_j,U_j^*)}{\partial U_j^*}\Bigg\}\\ &\quad +\delta_k^*\Bigg\{v_{kj}^*\frac{\partial f(S_k,S_k^*)}{\partial S_k^*}\cdot\frac{\partial f^*(U_j,U_j^*)}{\partial U_j^*}\\ &\quad +v_{kj}\frac{\partial f(S_k,S_k)}{\partial S_k}\cdot\frac{\partial f(U_j,U_j^*)}{\partial U_j^*}\Bigg\}\Bigg], \end{aligned} \tag{67}$$

$$\begin{aligned} \Delta v_{kj} &= \varepsilon H_j^*\Bigg[\delta_k^*\frac{\partial f(S_k,S_k^*)}{\partial S_k^*}\\ &\quad +\delta_k\frac{\partial f^*(S_k,S_k^*)}{\partial S_k^*}\Bigg]. \end{aligned} \tag{68}$$

双曲ＢＰ則 (式 (35), (36)) と複素ＢＰ則 (式 (67), (68)) を比較すると，次の点が異なっていることが分かる：

$$\begin{aligned} H_j &\longleftrightarrow H_j^*, && (69)\\ I_i &\longleftrightarrow I_i^*, && (70)\\ v_{kj} &\longleftrightarrow v_{kj}^*, && (71) \end{aligned}$$

$$\begin{aligned} &\delta_k\frac{\partial f(S_k,S_k^*)}{\partial S_k}+\delta_k^*\frac{\partial f^*(S_k,S_k^*)}{\partial S_k}\\ \longleftrightarrow\ &\delta_k^*\frac{\partial f(S_k,S_k^*)}{\partial S_k^*}+\delta_k\frac{\partial f^*(S_k,S_k^*)}{\partial S_k^*}. \end{aligned} \tag{72}$$

式 (69)〜式 (71) の違いは，$\nabla_{\boldsymbol{z}}$ と $\nabla_{\boldsymbol{z}^*}$ の違いから生じていると考えられる．また，式 (72) の違いは，それぞれの誤差関数の表現の違いから生じている．それは次のことから分かる．双

曲 BP 則の場合：

$$\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_n} = \frac{\partial}{\partial z_n}\frac{1}{2}\sum_{k=1}^{N}(\delta_k^2+\delta_k^{*2})$$
$$= -\frac{1}{2}\sum_{k=1}^{N}\left(\delta_k\frac{\partial O_k}{\partial z_n}+\delta_k^*\frac{\partial O_k^*}{\partial z_n}\right), \tag{73}$$

複素 BP 則の場合：

$$\frac{\partial E(\boldsymbol{z},\boldsymbol{z}^*)}{\partial z_n^*} = \frac{\partial}{\partial z_n^*}\frac{1}{2}\sum_{k=1}^{N}\delta_k\delta_k^*$$
$$= -\frac{1}{2}\sum_{k=1}^{N}\left(\delta_k^*\frac{\partial O_k}{\partial z_n^*}+\delta_k\frac{\partial O_k^*}{\partial z_n^*}\right). \tag{74}$$

5 結論

双曲関数に対する Wirtinger 微分を定義し，それを用いて双曲勾配オペレータを導出した．また，導いた双曲勾配オペレータを使って，いくつかの３層双曲 NN を対象にして，双曲誤差逆伝播学習アルゴリズムを導いた．誤差関数をパラメータの実部と Unipotent 部に関してそれぞれ偏微分して求めるこれまでの方法に比べると，双曲勾配オペレータを使う方法の計算の労力は半分で済む，学習則の表現が単純になり，プログラムのコーディングが容易になるといった利点がある．また，導いた双曲 BP 則と複素 BP 則の相違点について考察した．ただし，紙面の制約により，次の内容は省略した．(1) 双曲ＢＰ則 $\partial E/\partial z_k^*$ と複素ＢＰ則 $\partial E/\partial z_k^*$ の相違点，(2) 双曲ＢＰ則 $\partial E/\partial z_k^*$ と双曲ＢＰ則 $\partial E/\partial z_k$ の相違点，(3) 活性化関数の諸特性について．

謝辞 Wirtinger 微分に関する助言を下さった朝田衞教授（京都工芸繊維大学）に感謝致します．

参考文献

1) W. Wirtinger, “Zur Formalen Theorie der Funktionen von mehr Komplexen Veränderliehen,” Mathematische Annalen, Vol. 97, No. 1, 357/375 (1927).

2) D. H. Brandwood, “A Complex Gradient Operator and its Application in Adaptive Array Theory,” IEE Proc. F, Commun., Radar Signal Process., Vol. 130, No. 1, 11/16 (1983).

3) B. P. Palka, “An Introduction to Complex Function Theory,” Springer-Verlag (1995).

4) K. Kreutz-Delgado, “The Complex Gradient Operator and the CR-Calculus,” Lecture Supplement ECE275A, 1/74 (2006).

5) K. Kreutz-Delgado, “The Complex Gradient Operator and the CR-Calculus,” Available: http://arxiv.org/abs/0906.4835 (2009).

6) 高橋礼司，” 複素解析”，東京大学出版会 (1990).

7) D. P. Mandic, C. Jahanchahi and C. C. Took, “A Quaternion Gradient Operator and Its Applications,” IEEE Signal Processing Letters, Vol.18, No.1, 47/50 (2011).

8) M. F. Amin, M. I. Amin, A. Al-Nuaimi and K. Murase, “Wirtinger Calculus Based Gradient Descent and Levenberg-Marquardt Learning Algorithms in Complex-Valued Neural Networks,” Proc. Int. Conf. on Neural Information Processing (ICONIP2011), 550/559 (2011).

9) G. Sobczyk, “The Hyperbolic Number Plane,” The College Mathematics Journal, Vol.26, No.4, 268/280 (1995).

10) T. Nitta and S. Buchholz, “On the Decision Boundaries of Hyperbolic Neurons,” Proc. Int. Joint Conf. on Neural Networks (IJCNN2008), HongKong, 2973/2979 (2008).

11) S. Buchholz and G. Sommer, “A Hyperbolic Multilayer Perceptron,” Proc. Int. Joint Conf. on Neural Networks (IJCNN2000), Como, Italy, Vol.2, 129/133 (2000).

12) E. Hitzer, “Non-Constant Bounded Holomorphic Functions of Hyperbolic Numbers,” 第 1 回コンピューテショナル・インテリジェンス研究会講演論文集，23/28 (2011).

ニューロン生成／消滅アルゴリズムによる構造適応型 Restricted Boltzmann Machine

○鎌田　真（広島市立大学大学院情報科学研究科情報科学専攻）
市村　匠（県立広島大学経営情報学部経営情報学科）
原　章（広島市立大学大学院情報科学研究科）

A Proposal of Neuron Generation and Annihilation Method in Restricted Boltzmann Machine

∗S. Kamada (Graduate School of Information Sciences, Hiroshima City University)
T. Ichimura (Faculty of Management and Information Systems, Prefectural University of Hiroshima)
A. Hara (Graduate School of Information Sciences, Hiroshima City University)

Abstract– Restricted Boltzmann Machine (RBM) is a generative stochastic artificial neural network of energy-based model for unsupervised learning. Recently, RBM can be used as a method of pre-training in Deep Learning. In this paper, we will propose the adaptive RBM that can discover the optimal number of hidden neurons according to the training situation by applying the neuron generation and annihilation method. In this method, a new hidden neuron is generated when the fluctuation of the vector for the learning parameters is larger, and the inactivated hidden neurons are annihilated while the training. Some experiments in the proposed RBM were executed by using 2 kinds of data sets.

Key Words: Deep Learning, Restricted Boltzmann Machine, Neuron Generation and Annihilation

1　はじめに

近年の情報技術の発展に伴い，収集できる情報量は爆発的に増加し，これらの情報を扱うための多種多様な分析手法が提案されている．その中でも Deep Learning の有効性は次元圧縮，自然言語処理，音声処理，画像処理などの様々な分野で有効性が示されている [1)].

Deep Learning とは，階層型ニューラルネットワークの構造を持ち，実際の訓練の前に事前学習を行うことで，入力データの特徴を表す抽象概念を階層的に学習する機械学習手法である [2)]．従来の階層型ニューラルネットワークでは，多層にした場合，下位層において誤差が逆伝搬されにくく，過学習が生じる等の問題が生じるが，Deep Learning では，事前学習を行うことでこれらの問題を解決している．Restricted Boltzmann Machine(RBM)[3)] は尤度の概念を取り入れた統計的な手法であり，事前学習の一つの手法として知られている．Deep Belief Nets(DBN)[4)] は，RBM を階層的に積み上げ，各層に対して順次教師なし学習 (事前学習) を行い，その後パーセプトロンや誤差逆伝搬法などの教師あり学習を用いて微調整を行う学習手法である．

Deep Learning は多くの分野で優れた成果を残しつつも，最適なパラメタ設定は困難であることが知られている．例えば，RBM における隠れニューロン数や層の数は，学習に多くの影響を与える要素であり，これらの多くは分析者の経験により決められている．入力データに対して十分な隠れニューロン数や層の数がなければ，モデルは安定しない．従来の階層型ニューラルネットワークでは，学習中に更新される重みベクトルの振動の幅として Walking Distance(WD) と呼ばれる指標が定義され，これを用いて隠れニューロンの生成／消滅を行い，最適なネットワーク構造を自動で求める手法が提案されている [5, 6)]．一方，RBM ではエネルギー関数を用いてモデルの安定性が計算されているため，重みやエネルギーに係るパラメタがあり，これらが総合的な最適値をとることによって，RBM の学習が収束する．文献 [7)] では，RBM におけるエネルギーについて，リップシッツ連続を用いて，学習における各パラメタの変化量の上限が導出されており，これを満たすことで学習が収束していくことが示されている．そこで，本論文では，RBM の学習中におけるエネルギー関数と各パラメタの変分を観察し，入力データに対して最適なネットワーク構造を求めるニューロン生成／消滅アルゴリズムを提案する．手書き文字データセットとスマートフォンにより撮影された画像データを用いて提案手法の性能評価を行い，その結果をここに報告する．

本論文の構成は次のとおりである．2 節では RBM の基本的な仕組みについて述べる．3 節では，文献 [5, 6)] で提案されている階層型ニューラルネットワークにおける隠れニューロンの自動生成／消滅手法について述べ，4 節では，RBM における収束性について述べる．5 節で，本研究で提案する構造適応型 RBM について述べ，6 節で提案手法の性能評価を行った後，本論文をまとめる．

2　Restricted Boltzmann Machine

この節では，Rrestricted Boltzmann Machine(RBM) を説明する．RBM は，可視層と隠れ層の 2 層から構成され，確率分布に基づくネットワーク構造を持つモデルである．学習によって入力データを表現できるパラメタを獲得することができる．通常の Boltzmann Machine[9)] では，全ての層のニューロン間で相互結合があるが，RBM では，Fig. 1 のように，各層のニューロン間の結合を制限することで，計算量を削減するだけでなく，隠れ層の各ニューロンごとに独立した確率分布を学習す

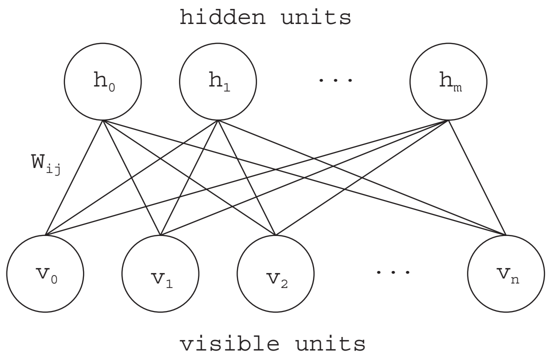


Fig. 1: Structure of RBM

ることができる．近年，RBM は Deep Learning における事前学習の 1 つの手法として使われ，特徴抽出の分野で注目されている．特に，RBM が Auto-Encoder[8] のような決定論的モデルとは異なり，エネルギー関数を用いる確率的なモデルであるため，学習したモデルの最適性を統計的な枠組みを用いて議論できる．

v_i を可視層における i 番目のニューロン，h_j を隠れ層における j 番目のニューロンとしたとき，エネルギー関数 $E(\boldsymbol{v}, \boldsymbol{h})$ は式 (1) のように定義され，また入力 $\boldsymbol{v}$ の尤度 $p(\boldsymbol{v})$ は式 (2) のように定義される．ここで，$\boldsymbol{v} \in \{0, 1\}^J$ と $\boldsymbol{h} \in \{0, 1\}^M$ はそれぞれ可視層と隠れ層のベクトルであり，J と M は可視ニューロン数，隠れニューロン数である．

$$E(\boldsymbol{v}, \boldsymbol{h}) = \sum_i b_i v_i - \sum_j c_j h_j - \sum_i \sum_j v_i W_{ij} h_j \quad (1)$$

$$p(\boldsymbol{v}, \boldsymbol{h}) = \frac{1}{Z} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) \quad (2)$$

$$Z = \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) \quad (3)$$

ここで，c_i は v_i に対するバイアス，b_j は h_j に対するバイアス，W_{ij} は v_i と h_j 間の重みである．また Z は分配関数 (partition function) であり，$\boldsymbol{v}$ と $\boldsymbol{h}$ の全ての可能な組み合わせを示す．

RBM におけるパラメタは，$\boldsymbol{v}$ の確率分布 $p(\boldsymbol{v}) = \sum_{\boldsymbol{h}} p(\boldsymbol{v}, \boldsymbol{h})$ の最尤推定により求められる．最尤推定は与えられた入力データから，尤もらしい確率分布を学習する方法であり，この場合，隠れ層のニューロンには相互結合がないため，各ニューロンごとに独立した確率分布を得ることができる．しかしながら，最尤推定における計算量は，モデルの大きさに応じて指数関数的に増加してしまい，現実的に計算困難である．そこで，マルコフ連鎖モンテカルロ法 (MCMC) に基づいたサンプリング方法として Contrastive Divergencde(CD) 法 [10] が提案され，現実的に計算可能となった．CD 法における最適なサンプリング回数については議論がされているが，サンプリング数が 1 回 (CD-1) でも十分な性能を示すことが知られている [11]．Fig. 2 は，サンプリング回数が 1 回の CD 法 (CD-1) のアルゴリズムを示している．

Step 1) 可視ベクトルに 1 つの訓練事例 $\boldsymbol{v}$ をセットする．

Step 2) すべての隠れニューロンに対して，可視ベクトル $\boldsymbol{v}$ を与えた際の条件付き確率 $p(h_j = 1|\boldsymbol{v})$ および状態変数 $h_j \in \{0, 1\}$ を計算する．

$$p(h_j = 1|\boldsymbol{v}) = sigm(b_j + \sum_i W_{ij} v_i) \quad (4)$$

ここで，$sigm()$ はシグモイド関数で $[0, 1]$ を出力する関数である．

Step 3) すべての可視ニューロンに対して，**Step 2)** で計算した隠れベクトル $\boldsymbol{h}$ を与えた際の条件付き確率 $p(v_i^{'} = 1|\boldsymbol{h})$ および状態変数 $v_i^{'} \in \{0, 1\}$ を計算する．

$$p(v_i^{'} = 1|\boldsymbol{h}) = sigm(c_i + \sum_j W_{ij} h_j) \quad (5)$$

Step 4) すべての隠れニューロンに対して，**Step 3)** で計算した可視ベクトル $\boldsymbol{v}^{'}$ を与えた際の条件付き確率 $p(h_j^{'} = 1|\boldsymbol{v}^{'})$ および状態変数 $h_j^{'} \in \{0, 1\}$ を計算する．

$$p(h_j^{'} = 1|\boldsymbol{v}^{'}) = sigm(b_j + \sum_i W_{ij} v_i^{'}) \quad (6)$$

Step 5) バイアスと重みを次のように更新する．

$$\begin{aligned} W_{ij} &= W_{ij} + \eta(v_i p(h_j = 1|\boldsymbol{v}) \\ &\quad - v_i^{'} p(h_j^{'} = 1|\boldsymbol{v}^{'})) \\ c_i &= c_i + \eta(v_i - v_i^{'}) \\ b_j &= b_j + \eta(p(h_j = 1|\boldsymbol{v}) \\ &\quad - p(h_j^{'} = 1|\boldsymbol{v}^{'})) \quad (7) \end{aligned}$$

ここで，η は学習率である．

Step 6) エネルギー関数が予め定められた値以下になる，または最大訓練回数に達するまで，**Step 1)** から **5)** までを繰り返す．

Fig. 2: Learning Algorithm of CD-1

3 階層型ニューラルネットワークにおける Walking Distance[5, 6]

従来の階層型ニューラルネットワークにおいて，学習中に更新される重みベクトルの振動を用いて，隠れニューロン数を動的に変更する手法が提案されている．入力データに対して十分な隠れニューロン数があれば，Fig. 3 のように，学習が進むにつれて，重みベクトルの振動の幅は小さくなり，ある一定の値に収束する傾向があると考えられている．一方で，十分な隠れニューロン数がなければ，学習がある一定以上進んでも，重みベクトルの振動の幅は大きく，収束しないと考えられる．この場合，最も振動している隠れニューロンの重

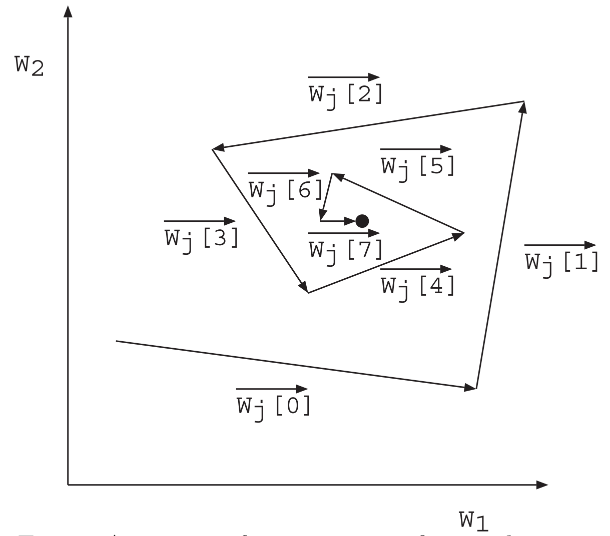


Fig. 3: An image of convergence of a weight vector

みを 2 つに分割し，新しい隠れニューロンを追加することで振動の幅は小さくなり，モデルが安定していく．式 (8) は，ある隠れニューロン j において，訓練回数 n までの重みベクトルの振動の和を示しており，Walking Distance と呼ばれる．

$$WD_j[n] = \sum_{m=1}^{n} Met(\vec{W}_j[m], \vec{W}_j[m-1]) \tag{8}$$

ここで，$\vec{W}_j[m]$ は，隠れニューロン j における重みベクトルであり，Met はベクトル間の距離を計算する関数で，ここではユークリッド距離が使われる．文献[5, 6] では，ある一定回数の訓練後，隠れニューロン j の Walking Distance である WD_j の変化量が予め定められた閾値よりも大きい場合，その隠れニューロンの重みを 2 分割し，新しく隠れニューロンを追加する手法が提案されている．

$$\Delta\varepsilon_j = \frac{\partial\varepsilon}{\partial WD_j} \cdot WD_j > \theta_G \tag{9}$$

ここで，ε はネットワーク全体の誤差であり，θ_G は閾値である．

一方，入力データに対して十分な数の隠れニューロンがある場合，ネットワークの出力に寄与していない隠れニューロンが存在する場合がある．この場合，ネットワークの構造が複雑になり，計算量が多くなる．文献[5, 6] では，活性化していない隠れニューロン，すなわち全ての入力データに対して常に 0 を出力する隠れニューロンを出力に寄与していないとみなし，その隠れニューロンを削除する手法が提案されている．

4　RBM の CD 法における収束性[7]

2 節で述べたように，一般的に RBM の学習には CD 法が使われている．CD 法は少ないサンプリング回数でモデルの確率分布を推定できるアルゴリズムであるが，Fig. 2 における Step 2)，Step 3)，Step 4) のサンプリングの過程で，各ニューロンが出力する実数の条件付き確率が二値に変換され，誤差が生じる．その結果，Step 5) の勾配計算において，誤差が生じたままパラメタが更新され，モデルが安定せず，エネルギーが収束しない場合がある．

文献[7] では，RBM の分配関数が凸関数であることを仮定し，リップシッツ連続を用いて学習における勾配の上限を理論的に導出している．ここでは，入力データが観測されたときの対数尤度について，あるパラメタ $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W}\}$ に対する勾配を解析するために，可視ベクトル $\boldsymbol{v}$ と隠れベクトル $\boldsymbol{h}$ のエネルギーが式 (10) のように定義されている．

$$-E(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta}) = \sum_i v_i b_i + \sum_j h_j c_j + \sum_i \sum_j v_i W_{ij} h_j \tag{10}$$

$$\begin{aligned} \arg\min_{\boldsymbol{\theta}} F(\boldsymbol{\theta}) &= \frac{-1}{N} \sum_{n=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{v}_n) \\ &= f(\boldsymbol{\theta}) - g(\boldsymbol{\theta}) \end{aligned} \tag{11}$$

$$f(\boldsymbol{\theta}) = \log \sum_{\boldsymbol{v}} \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta})) \tag{12}$$

$$g(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \log \sum_{\boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h}; \boldsymbol{\theta})) \tag{13}$$

ここで観測データは $\boldsymbol{v}_n = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_N\}$ と定義され，N はデータ数である．式 (11) は目的関数である対数尤度を示しており，式 (12)，式 (13) は，それぞれモデルに対する尤度と入力データに対する尤度を示しており，特に $f(\boldsymbol{\theta})$ は分配関数と呼ばれる．文献[7] では $f(\boldsymbol{\theta})$ が凸関数であることを仮定し，テイラー展開を用いて分配関数の各パラメタの勾配に対する上限が式 (14)，式 (15)，式 (16) のように導出されている．

$$f(\{\boldsymbol{b}, \boldsymbol{c}^k, \boldsymbol{W}^k\}) \leq f(\boldsymbol{\theta}^k) + \langle \nabla_{\boldsymbol{b}} f(\boldsymbol{\theta}^k), \boldsymbol{b} - \boldsymbol{b}^k \rangle + \frac{J}{2} \|\boldsymbol{b} - \boldsymbol{b}^k\|_\infty^2 \tag{14}$$

$$f(\{\boldsymbol{b}^k, \boldsymbol{c}, \boldsymbol{W}^k\}) \leq f(\boldsymbol{\theta}^k) + \langle \nabla_{\boldsymbol{c}} f(\boldsymbol{\theta}^k), \boldsymbol{c} - \boldsymbol{c}^k \rangle + \frac{M}{2} \|\boldsymbol{c} - \boldsymbol{c}^k\|_\infty^2 \tag{15}$$

$$\begin{aligned} f(\{\boldsymbol{b}^k, \boldsymbol{c}^k, \boldsymbol{W}\}) \leq\ & f(\boldsymbol{\theta}^k) + tr((\boldsymbol{W} - \boldsymbol{W}^k)^T \nabla_{\boldsymbol{W}} f(\boldsymbol{\theta}^k)) \\ & + \frac{2MJ}{2} \|\boldsymbol{W} - \boldsymbol{W}^k\|_{S^\infty}^2 \end{aligned} \tag{16}$$

ここで，J，M はそれぞれ可視ニューロン数，隠れニューロン数である．S^∞ はシャッテンノルムである．$\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ はベクトルの内積である．各式の右辺の第 2 項は 1 次の偏微分の勾配であり，第 3 項はステップ幅である．これらの式は，リップシッツ連続における上限式と同じ意味を持ち，すなわち学習によって各パラメタの勾配が徐々に小さくなっていき，学習が収束することを示している．文献[5, 6] の階層型ニューラルネットワークでは，重みの勾配のみを小さくすることで学習が収束することが示されているが，以上のことから，RBM における CD 法の学習では，重みの勾配だけでなく，可視ニューロンと隠れニューロンのパラメタの勾配をもとに学習が上手く行われているかどうかを判断する必要がある．

5　構造適応型 RBM

3 節，4 節で述べたように，階層型ニューラルネットワークでは重みの勾配ベクトルを小さくすることで学習が収束するが，RBM では，3 つパラメタ $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W}\}$

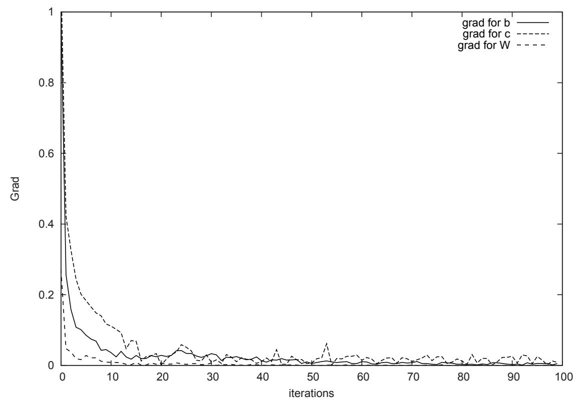


Fig. 4: Grad for b, c and W

の勾配が徐々に小さくなることで学習が収束する．予備実験において，“MNIST[12]” と呼ばれる手書き文字データセットを用いて各パラメタの勾配を調査したところ，Fig. 4 のような結果が得られた．Fig. 4 は，隠れニューロン数が 10 個の場合，学習中の $\boldsymbol{b}, \boldsymbol{c}, \boldsymbol{W}$ の勾配を示している．実験結果から，各パラメタの勾配は学習が進むにつれて徐々に小さくなったが，その中でも，隠れニューロンに関する $\boldsymbol{c}$ の勾配の変化量が多いことが分かった．一方，可視ニューロンに関する $\boldsymbol{b}$ は入力に対するパラメタであるため，入力データの分布に応じて変動が異なると考えられる．そこで，本論文では可視ニューロンの勾配ベクトルは考慮せず，隠れニューロンの勾配ベクトルと重みの勾配ベクトルを考慮したベクトルを考え，この変化量があらかじめ定められた閾値を超えたとき，隠れニューロンを挿入する手法として構造適応型 RBM を提案する．

$$(\alpha_c \cdot dc_j) \cdot (\alpha_W \cdot dW_{ij}) > \theta_G \tag{17}$$

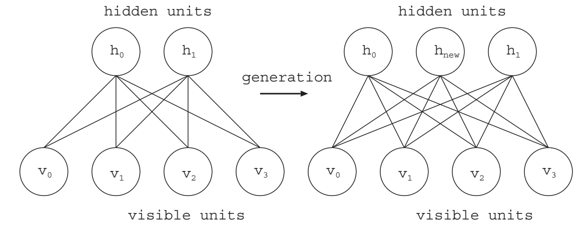
ここで，dc_j，dW_{ij} は，それぞれ j 番目の隠れニューロンのパラメタの勾配，i, j 番目の重みの勾配を示している．α_c，α_W は定数であり，各パラメタの勾配のスケールを調整するために使われる．θ_G は閾値である．式 (17) を満たす隠れニューロンは，Fig. 5(a) のように，隠れ層の中の該当する位置にニューロンが挿入される．このとき，挿入される隠れニューロンのパラメタおよび重みの初期値は，挿入位置のニューロンのパラメタ値のコピーとする．

一方，入力データに対して十分な数の隠れニューロンがある場合，出力に寄与していない隠れニューロンが存在する場合がある．そこで，一定の訓練後，全ての入力データに対して，式 (4) における隠れニューロンの出力値の平均がしきい値を下回った場合，該当する隠れニューロンを Fig. 5(b) のように消滅させる．

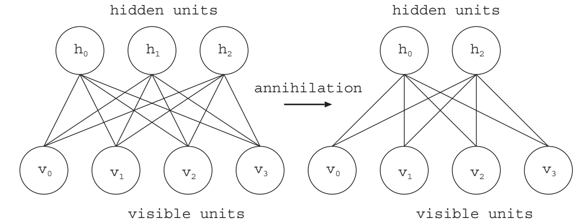
$$\frac{1}{N}\sum_{n=1}^{N} p(h_j = 1|\boldsymbol{v_n}) < \theta_A \tag{18}$$

ここで，N は入力データ数であり，θ_A はしきい値である．

提案する構造適応型 RBM では，学習中のある時点において，式 (17) を満たす隠れニューロンが存在すれば，その隠れニューロンを 1 個選択し，選択したニューロンの属性値を継承した新しい隠れニューロンを 1 個



(a) Neuron Generation



(b) Neuron Annihilation

Fig. 5: Adaptive RBM

生成し，その近傍に挿入する．その後，一定の訓練を行い，式 (18) を満たす隠れニューロンが存在すれば，ニューロンの消滅を行う．

6 実験

6.1 データセット

本節では，提案する構造適応型 RBM を “MNIST[12]” と，スマートフォンアプリケーション “ひろしま観光マップ [13]” により収集された旅行者の写真のデータセットに適用し，アルゴリズムの有効性を検証した．MNIST は数字の 0 から 9 の手書き文字画像のデータセットであり，訓練事例 60,000 枚，テスト事例 10,000 枚から構成される．各画像は 28×28 ピクセルである．ひろしま観光マップは，本研究において開発した MPPS アプリケーションであり，観光地の訪れた旅行者の主観的情報として位置情報，5 段階評価値，コメント文，写真を収集することができる [14]．ここでは，収集された写真を用いて，「宮島」(Fig. 6(a)),「原爆ドーム」(Fig. 6(c)),「戦艦大和」(Fig. 6(e)) の 3 つのクラスを用いた．各クラスに対して 288 枚の訓練画像と 32 枚のテスト画像があり，各画像は 48×48 ピクセルである．

RBM の実装には，Pylearn2[15] と呼ばれる Deep Learning のライブラリを用いた．学習アルゴリズムには Stochastic Gradient Descent (SGD) を用い，学習のバッチサイズは 100 とした．また実験には次のスペックを持つ PC を用いた．CPU = Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz, GPU = GeForce GTX 960, Memory = 16GB, OS = Fedora 22 64 bit.

6.2 実験結果

Fig. 7 は，MNIST に対して，隠れニューロンの初期値を 10 個，$\theta_G = 0.005$，$\theta_G = 0.15$ に設定したとき，RBM のエネルギー，各パラメタの勾配，隠れニューロン数について従来の RBM と提案手法の RBM を比較した結果を示している．Fig. 8 は隠れニューロンの初期値を 100 個，$\theta_G = 0.03$，$\theta_G = 0.3$ にした場合の結果である．隠れニューロンの初期値が 10 個の場合，Fig. 7(a) に示すように，従来の RBM では，エネルギーの値が小さくならず，安定していない箇所が見られたが，提案手

(a) Torii in Miyajima 1 (b) Torii in Miyajima 2

(c) Atomic Bomb Dome 1 (d) Atomic Bomb Dome 2

(e) Battleship Yamato 1 (f) Battleship Yamato 2

Fig. 6: An example of Hiroshima Tourist Map

法の RBM では，学習が進むにつれてエネルギーが徐々に小さくなり，安定した．これは，Fig. 7(b) において，可視層のパラメタの勾配が上昇し，隠れニューロンが生成されたためである．ニューロン生成の直後はエネルギーや各パラメタの勾配は一時的に振動したが，再度学習した結果，これらは安定し，学習後期にはエネルギーだけでなく全てのパラメタの勾配を従来の RBM よりも小さくすることができた (Fig. 7(b)，Fig. 7(c))．最終的に，隠れニューロン数は 10 個から 61 個になった (Fig. 7(d))．

一方，Fig. 8 に示すように，隠れニューロンの初期値が 100 個の場合は，提案手法の RBM の隠れニューロン数は消滅した．Fig. 8(d) に示すように，最終的に隠れニューロン数は 100 個から 72 個になった．ニューロン生成時と同様に，消滅の直後はエメルギーと各パラメタの勾配は一時的に不安定になったが，学習が進むことでこれらは安定し，提案手法の RBM において隠れニューロンを削除してもモデルが不安定にならないことが分かった (Fig. 8(a)，Fig. 8(b)，Fig. 8(c))．

Fig. 9 は，ひろしま観光マップのデータに対して，隠れニューロン数の初期値を 100 個，$\theta_G = 0.015$，$\theta_G = 0.01$ にした場合の実験結果であり，Fig. 10 は，同じデータに対して，隠れニューロン数の初期値を 300 個，$\theta_G = 0.035$，$\theta_G = 0.06$ にした場合の実験結果を示している．全体的に，ひろしま観光マップの結果は MNIST と比べるとエネルギーや勾配の振動が大きくなったが，従来の RBM と提案手法の RBM の実験結果については MNIST と同様な傾向が見られた．つまり，隠れニューロン数が少ない場合はニューロン生成過程が適用され，エネルギーは安定したが，十分な隠れニューロン数がある場合は，モデルが不安定にならない程度においてニューロン消滅過程が適用された．最終的に，隠れニューロン数の初期値が 100 個の場合は，229 個になり，隠れニューロン数の初期値が 300 個の場合は 240 個になった．また，隠れニューロン数の初期値が 300 個場合，学習後の RBM の構造に出力層を加え，パーセプトロンによる識別を行ったところ，従来の RBM では訓練データに対して 97.1%の識別率，テストデータに対して 88.5%の識別率を示したが，提案手法の RBM でも正答率に違いは見られず，提案手法の RBM の有効性を示すことができた．

7 おわりに

Deep Learning と呼ばれる機械学習の手法が様々な分野で高い精度を示しており，現実世界の問題に応用されている．Deep Learning の事前学習には，Auto-Encoder と RBM から派生するアルゴリズムがよく利用されている．Auto-Encoder に比べて，RBM は尤度の概念を用いた統計的なモデルであるため，優れた手法であると言われているが，実際には構造の複雑さなどの理由から，Auto-Encoder を積み重ねた Stacked Auto-Encoder や CNN などの手法の方がよく利用されている．本研究では，RBM に着目し，CD 法を使った学習の収束性について述べるとともに，学習中にニューロンの生成／消滅を行うことで最適な構造を求める構造適応型 RBM を提案した．いくつかのデータ・セットに適用したところ，高い有効性を持つことがわかった．今後は，他のデータセットに対しても適用し，有効性を検証するとともに，構造適応型 RBM の学習により獲得された構造から，IF-THEN ルールのような明確な知識構造を獲得する．

謝辞

本研究は JSPS 科研費 25330366 と平成２７年度経済産業省戦略的基盤技術高度化支援事業の助成の一部を受けたものである．

参考文献

1) V.Le.Quoc, R.Marc's Aurelio, et.al: *Building high-level features using large scale unsupervised learning*, International Conference in Machine Learning (2012)

2) Y.Bengio: *Learning Deep Architectures for AI*, Foundations and Trends in Machine Learning archive, Vol.2, No.1, 1/127 (2009)

3) G.E.Hinton: *A Practical Guide to Training Restricted Boltzmann Machines*, Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, Vol.7700, 599/619 (2012)

4) G.E.Hinton, S.Osindero and Y.Teh: *A fast learning algorithm for deep belief nets*, Neural Computation, Vol.18, No.7, 1527/1554 (2006)

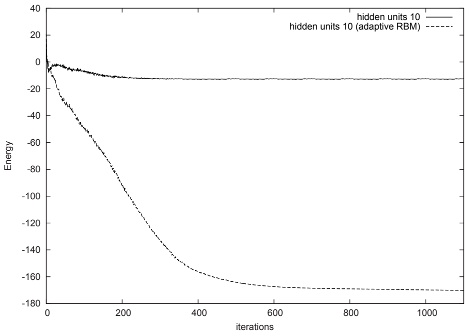
5) T.Ichimura: *Studies on Learing and Reasoning Methods in Neural Networks*, Ph.D. Thesis, Toin University of Yokohama (1997)

6) T.Ichimura, and K.Yoshida Eds.: *Knowledge-Based Intelligent Systems for Health Care*, Advanced Knowledge International (ISBN 0-9751004-4-0) (2004)

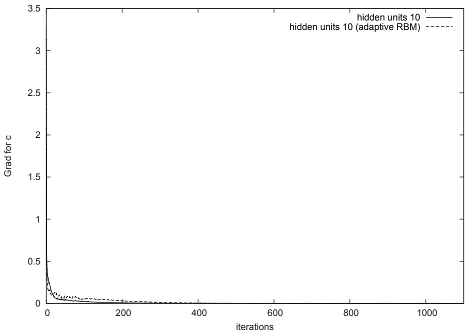
7) D.Carlson, V.Cevher and L.Carin: *Stochastic Spectral Descent for Restricted Boltzmann Machines*, Proc. of the Eighteenth International Conference on Artificial Intelligence and Statistics, 111/119 (2015)

8) Y.Bengio, P.Lamblin, D.Popovici and H.Larochelle: *Greedy Layer-Wise Training of Deep Networks*, in Advances in Neural Information Processing Systems 19 (NIPS ‘ 06), 153/60 (2007)

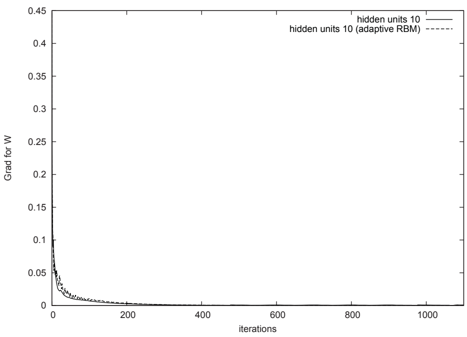
9) D.H.Ackley, G.E.Hinton and T.J,Sejnowski: *A Learning Algorithm for Boltzmann Machines*, Cognitive Science, 9: 147/169. doi: 10.1207/s15516709cog0901_7 (1985)



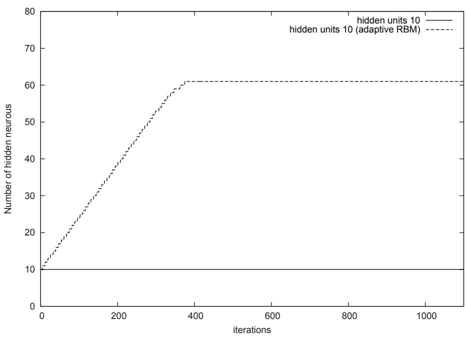
(a) Energy Function



(b) Grad for c

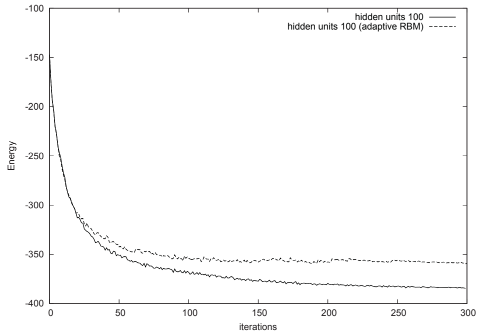


(c) Grad for W

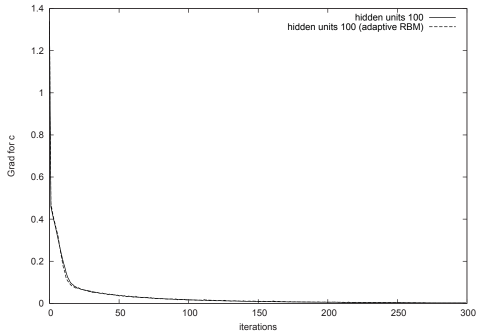


(d) Number of hidden neurous

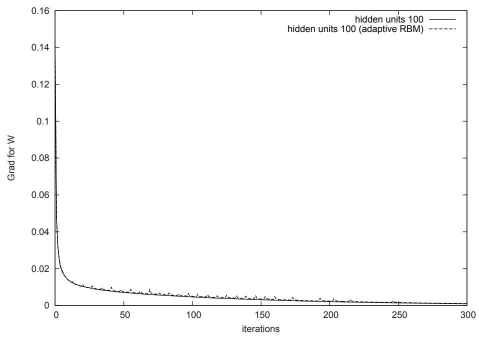
Fig. 7: MNIST (initial hidden neurous = 10)



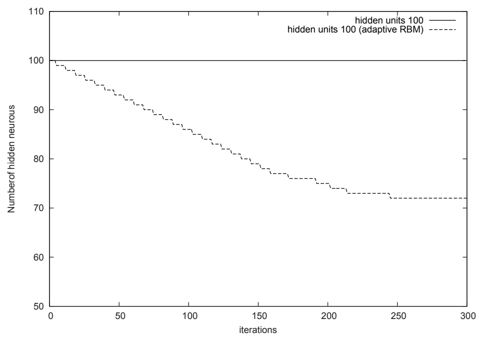
(a) Energy Function



(b) Grad for c



(c) Grad for W



(d) Number of hidden neurous

Fig. 8: MNIST (initial hidden neurous = 100)

10) G.E.Hinton: *Training products of experts by minimizing contrastive divergence*, Neural Computation, Vol.14, 1771/1800 (2002)

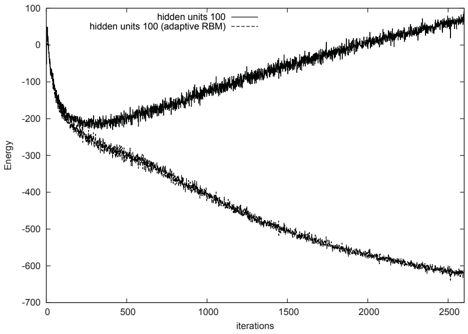
11) T.Tieleman: *Training restricted Boltzmann machines using approximations to the likelihood gradient*, Proc. of the 25th international conference on Machine learning, 1064/1071 (2008)

12) Y.LeCun, et.al: *THE MNIST DATABASE of handwritten digits*, `http://yann.lecun.com/exdb/mnist/`, [online] (2015/12/2 閲覧)

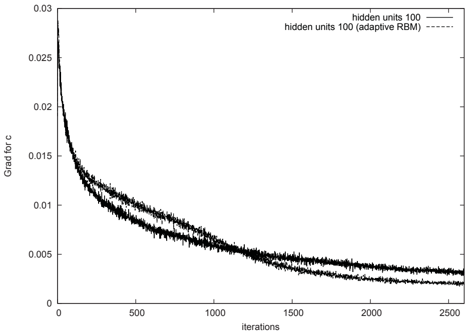
13) ITProducts, ひろしま観光マップ: `https://play.google.com/store/apps/details?id=jp.itproducts.KankouMap`, [online] (2011)

14) T.Ichimura, S.Kamada and K.Kato: *Knowledge Discovery of Tourist Subjective Data in Smartphone Based Participatory Sensing System by Interactive Growing Hierarchical SOM and C4.5*, Intl. J. Knowledge and Web Intelligence, Vol.3, No.2, 110/129 (2012)

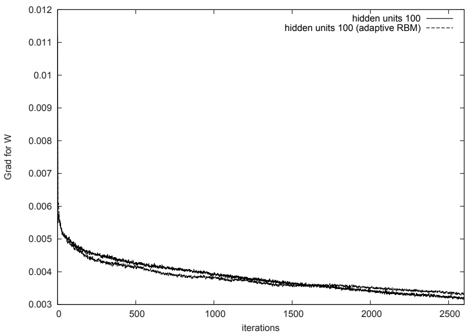
15) I.Goodfellow, David Warde-Farley, et.al.: *Pylearn2:*



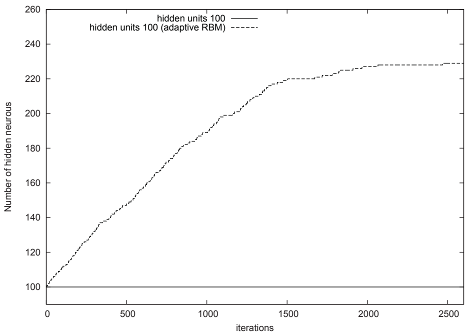
(a) Energy Function



(b) Grad for c



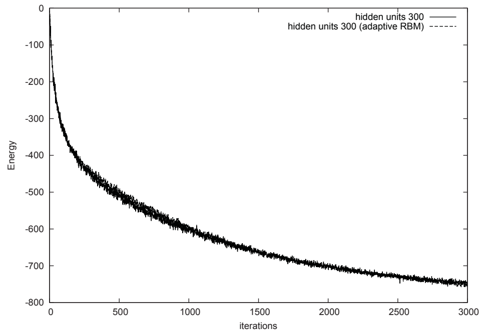
(c) Grad for W



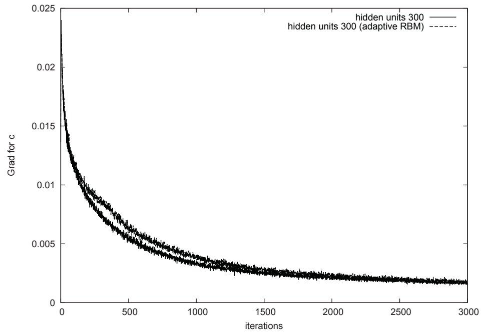
(d) Number of hidden neurous

Fig. 9: Hiroshima Tourist Map (initial hidden neurous = 100)

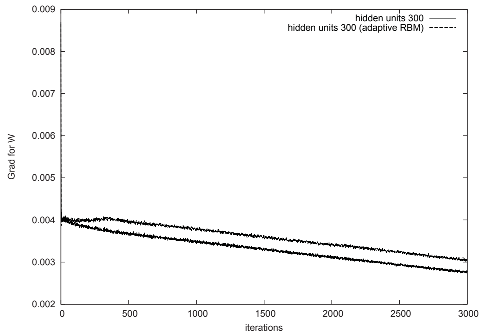
a machine learning research library, arXiv preprint arXiv:1308.4214 (2013)



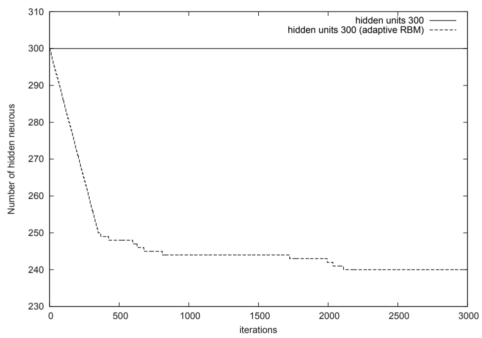
(a) Energy Function



(b) Grad for c



(c) Grad for W



(d) Number of hidden neurous

Fig. 10: Hiroshima Tourist Map (initial hidden neurous = 300)

深層学習による教師なし画像の近傍空間の学習

○西先崇　濱津文哉　濱上知樹（横浜国立大学）

Learning neighborhood image space of unlabeled images with deep learning

∗T. Nishisaki, F. Hamatsu, and T. Hamagami (Yokohama National University)

Abstract– A method of learning neighborhood image space of unlabeled images with deep learning is proposed. For classification problem of the large image dataset, labeled data is indispensable property. However, preparing sufficient labeled images costs more, compared to the unlabeled images. In order to overcome the problem, a method for classification of unlabeled images with a few labeled image and interpolation of them is examined. The proposed method learns the neighborhood image space of the representative images for target class with Google Similar Image Retrieval. Experiment results show it is confirmed that the proposed method is effective for image classification of the target class with only unlabeled images.

Key Words: Deep learning, Image classification, Unlabeled images

1　はじめに

近年, 高度な画像分類を用いた様々なアプリケーションが開発され，その学習精度の向上がさらに期待されている．画像分類学習では，分類カテゴリのラベルを有する教師あり画像を用いて，未知の画像に対する画像分類を行うのが一般的である．

しかし，教師あり画像のサンプル数が十分でない場合，画像空間全体の学習が不完全となり，未知の画像の分類に失敗する．例えば，ラベル付与に専門家の知識が必要な医療や人文系のデータでは，十分な数の教師データがそろわず，大規模な学習は困難である．

この問題に対し，本研究では，教師（ラベル）なし画像の近傍空間を用いた学習方法を提案する．本手法では，分類カテゴリに属する教師なし代表画像をもとに，一般画像空間中における近傍サンプルを収集し，これを用いて近傍空間の学習を行う．この方法を用いることにより，教師なし画像から学習に必要なクラスタを構成し，クラスタの特徴を抽出することで分類を可能にする．

2　画像分類

一般に教師あり画像群を用いた画像分類では，画像をある写像関数 F によって画像特徴空間に写像して識別面を学習する．教師あり画像群を用いた分類を Fig. 1 に示す．画像特徴空間へ写像し画像特徴量を得る手法として SIFT（Scale-Invariant Feature Transform）や SURF（Speeded Up Robust Features）などが知られている．得られた画像特徴量を用い，教師あり画像群の持つラベル情報に従って学習することで，目的とするカテゴリ分類器が得られる．近年では画像特徴量の抽出と分類を共に学習する手法として深層学習 [1)2)] が注目され，人手に依らない高度な特徴量抽出により高い画像分類精度を達成している．

しかし，教師あり画像を十分な数用意することが困難な場合, カテゴリの分類は困難になる．このような場合，教師なし学習である k-means 法や自己組織化マップ (Self Organizing Map: SOM) を用いることにより，特徴空間中でクラスタをつくり，これをカテゴリと対応づける方法が考えられる．しかし，この方法は分類すべきカテゴリとクラスタが 1 対 1 になるとは限らない．また，クラスタを構成するサンプルが十分そろわず，特徴空間に対してスパースな場合は，クラスタを構成することが困難である．特に深層学習のように，特徴ベクトルの学習に多くのデータを必要とする場合は，大きな問題になる．

そこで本研究では，スパースなデータ空間にある少数の教師あり画像から，その近傍空間を代表する特徴ベクトルを深層学習によって得るための方法について扱う．具体的には，カテゴリに対応した教師あり画像をもとに一般画像空間から Google 画像検索 [3)] を用いて類似画像を収集し，疑似的につくられた画像クラスタを用いて深層学習を行うことで，カテゴリに対応した特徴ベクトルを抽出する．

3　深層学習

深層学習は従来のニューラルネットワークよりも多段の層を持ち，様々な分野で高い性能を達成している手法である．深層学習は目的タスクの性質などから様々な提案がされているが，本稿では画像認識において最も広く用いられている畳み込みニューラルネットワーク [4)5)] を用いる．

畳み込みニューラルネットワークは局所領域に対して重みを畳み込む処理を行う層を持つ順伝播型ネットワークであり，教師ありデータを用いて逆誤差伝播法により学習を行う．Fig. 3 に示すように，局所領域に対して重みの畳み込み計算を行うことで，生物の脳の視覚受容野を模倣しており，画像認識において重要な技術となっている．画像認識でよく用いられる畳み込みニューラルネットワークは，畳み込み層とプーリング層を交互に接続しその後全結合層が続き，最後に各カテゴリの出力を持つ．

単一チャネルを持つ画像について，各画素を $x_{i,j}$ とする．また畳み込み処理を行う際の重みの値の集合をフィルタと呼び，そのフィルタのサイズを $H \times H$ とし各フィルタの値を $h_{p,q}$，バイアスを $b_{i,j}$，出力を $u_{i,j}$ とすると，画像の畳み込みは式 (1) で示される．

$$u_{i,j} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{p,q} + b_{i,j} \tag{1}$$

式 (1) では縦横方向に 1 画素ずつずらしながら畳み込みを計算しているが，このずらす幅をストライドと

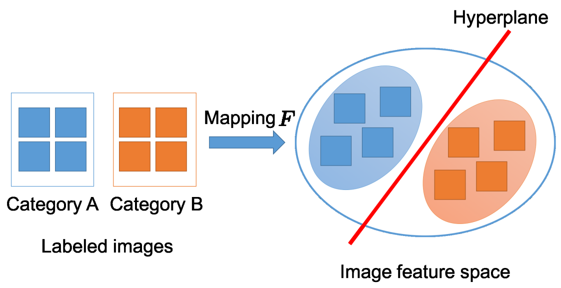


Fig. 1: Classification with labeled images.

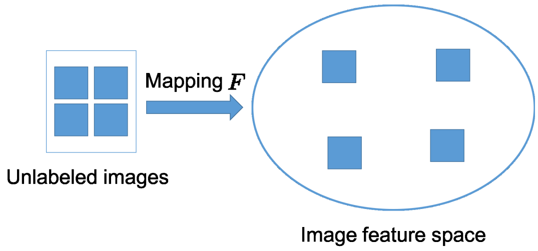


Fig. 2: Classification with unlabeled images.

呼び，場合によってはストライドを数画素ずつずらすことがある．ストライドを s とすると，ストライドを考慮した畳み込みは式 (2) で示され，出力画像サイズは約 $1/s$ 倍となる．

$$u_{i,j} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{si+p,sj+q} h_{p,q} + b_{i,j} \tag{2}$$

カラー画像のような複数チャネルの画像に対しては各チャネルごとに同様の畳み込み処理を行い，また畳み込むフィルタ数も複数チャネルにする場合がある．入力が K チャネルの画像の時，入力チャネル k，フィルタ m に対する畳み込みは式 (3) で示される．またバイアス $b_{i,j,m}$ はフィルタごとに共通の値を持つのが一般的である．

$$u_{i,j,m} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q,k} h_{p,q,k,m} + b_{i,j,m} \tag{3}$$

畳み込みニューラルネットワークにおける各フィルタの値は同一チャネルの全画素で同じ値を用いており，これは共有重みと呼ばれている．

畳み込みによって得られた $u_{i,j,m}$ に対し，活性化関数を適用することで最終的な出力 $z_{i,j,m}$ が得られ，式 (4) で示される．

$$z_{i,j,m} = f(u_{i,j,m}) \tag{4}$$

様々な活性化関数が提案されているが，近年では式 (5) に示される正規化線形関数 (rectified linear function: ReL) が一般的に用いられる．正規化線形関数は活性化関数の中でも計算量が少なく，学習の収束が速いことが知られている．

$$f(u) = \max(u, 0) \tag{5}$$

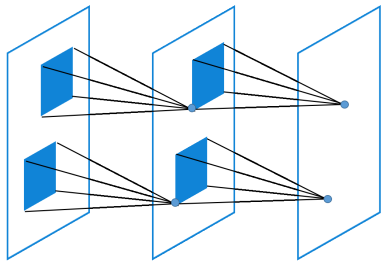


Fig. 3: Convolution.

畳み込み層の後に繋げるプーリング層では，畳み込み層で得られた出力の中で一部の情報を間引く処理を行うことで画素の位置変化にロバストな特徴を得ている．プーリングには様々な種類があるが，画像認識においては最大プーリング（Max-pooling）が一般的であり，本稿においてもこの手法をとる．プーリングを行う領域のサイズを $H \times H$ としたとき，最大プーリングでは H^2 個の画素値の中で最大値以外の値を間引く処理を行う．

4 近傍空間の学習

本研究ではスパースなデータ空間にある少数の教師あり画像から画像分類を行うため，分類カテゴリに属する教師あり画像を各カテゴリでそれぞれ 1 枚選択し，深層学習により近傍空間を形成する．提案手法における近傍空間の学習の概略図を Fig. 4 に示す．

スパースなデータ空間において画像空間全体の学習を行うのが困難であるという課題を解決するため，提案手法では一般画像空間上で代表画像と類似した画像群を収集し学習を行う．これによりスパースであったデータ空間中のサンプルを補完し，画像空間全体の学習の不完全性を緩和することが期待できる．

提案手法による画像分類の概略図を Fig. 5 に示す．提案手法では学習フェーズと分類フェーズの 2 つのフェーズから成る．

学習フェーズでは，教師なし画像群から各カテゴリに属する代表画像を選択し，Google 画像検索により画像特徴空間上で類似した画像群を収集する．Google 画像検索によって得られた画像群を用いて学習に必要なクラスタを構成し，得られた特徴量による分類を行う．クラスタの構成，特徴量の抽出は画像認識で広く用いられる深層学習として畳み込みニューラルネットワークを用いる．

分類フェーズでは，未知の画像に対し学習フェーズで構築した畳み込みニューラルネットワークを用いて特徴量を抽出し，カテゴリの分類を行う．

5 実験

提案手法の有効性を確認するため，以下の 2 つの実験を行った．

5.1 実験設定

1 つ目の実験では，画像特徴空間上の近傍空間の学習精度を確認するため，訓練データの一部をバリデーションデータとして分離し，バリデーションデータに対する分類精度を求めた．各カテゴリの代表画像をクエリとして得られる Google 画像検索の結果のうち検索結果順に 300 枚の画像を収集し，そのうち 90 % を占

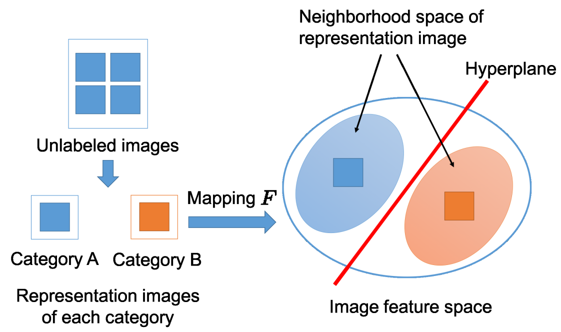


Fig. 4: Classification with proposed method.

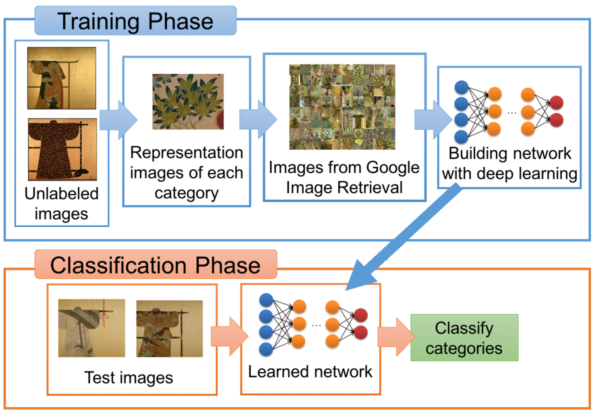


Fig. 5: Overview of proposed method.

める 270 枚の画像を訓練画像，残りの 10 %を占める 30 枚の画像をバリデーションデータとして学習し分類した．

2 つ目の実験では，提案手法による画像分類の精度を確認するため，学習済みの畳み込みニューラルネットワークを用いて画像分類を行った．比較する従来手法として教師あり画像群のサンプル数が少ない場合を想定し，各カテゴリの代表画像を学習データとした．また各カテゴリに分類された画像を確認することで誤分類した場合の原因について考察した．

本実験では画像データセットとして国立歴史民俗博物館蔵の小袖屏風画像 [6] を用いた．小袖屏風画像については 5.2 節で述べる．

本実験で分類するカテゴリとして，館蔵野村正治郎衣裳コレクションデータベース [7] に記載されているモチーフ（模様や柄）を用いた．代表画像は 50 種類のモチーフについて，小袖屏風画像を分割した画像群から人為的に選択し，80×80 画素のカラー画像となっている．小袖屏風分割画像群は 1 隻の小袖屏風画像を縦に 40，横に 30 分割して得られた 1200 枚の画像群に対し，80×80 画素のサイズに満たない切れ端の画像を省き，1 隻の小袖屏風画像に対し 1141 枚得られた．小袖屏風分割画像群は 102 隻の小袖屏風画像から得られた．学習に用いたモチーフの代表画像の例を Fig. 7 に示す．2 つ目の実験におけるテストデータは各モチーフの代表画像を含む 276 枚の小袖屏風分割画像である．

実験に用いた深層学習のネットワークは Table 1 の通りである．

Table 1: Network structure.

Name	Patch	Stride	Output map size	Activation function
Input	-	-	$80 \times 80 \times 3$	-
Convolution1	5×5	1	$76 \times 76 \times 15$	ReL
Max-pooling1	2×2	1	$38 \times 38 \times 15$	-
Convolution2	5×5	1	$34 \times 34 \times 30$	ReL
Max-pooling2	2×2	1	$17 \times 17 \times 30$	-
Fully-connected1	-	-	$1 \times 1 \times 8000$	ReL
Fully-connected2	-	-	$1 \times 1 \times 50$	-

Fig. 6: Kosode byobu images（小袖屏風画像）.

Fig. 7: Examples of representation images.

5.2 小袖屏風

小袖屏風画像は教師あり画像群の確保が困難な例として挙げられるデジタルアーカイブの一種である．デジタルアーカイブは博物館，美術館などの文化資源をデジタル化して保存したデータであるが，文化資源の数は有限であるためにデータ数が限られ，描かれるモチーフの分類は専門家でなければ難しい．

小袖屏風は日本の伝統的衣装である小袖を屏風に貼り付けた歴史資料であり，Fig. 6 に例を示す．

6 結果と考察

6.1 近傍空間の学習に関する結果

近傍空間の学習精度を評価するため，Google 画像検索によって得られた訓練データの一部をバリデーションデータとして分離し学習を行った．分類精度は 50 モチーフに対する平均値とした．結果を Fig. 8 に示す．結果から，学習が進むにつれて訓練データに対する分類精度は向上するが，バリデーションデータに対する精度は 35 %程度で収束したことが明らかとなった．訓練データとバリデーションデータの精度の差が生じたため過学習が発生し，差が 50 %以上開いたことから，

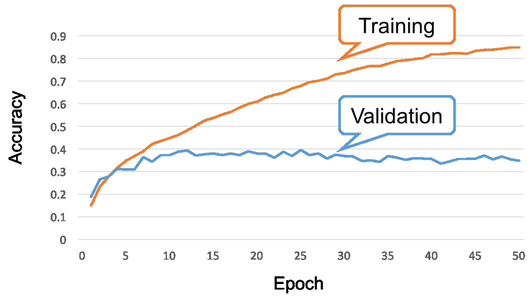


Fig. 8: Accuracy of neighborhood space.

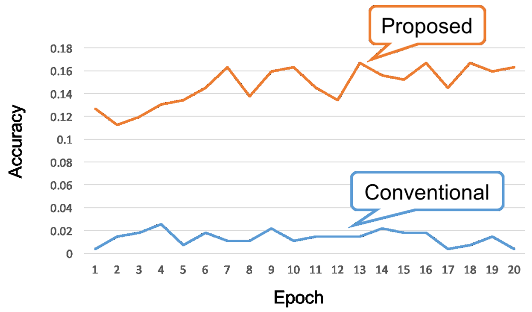


Fig. 9: Accuracy of motif classification.

汎化性能が不十分であることがわかる．また，本手法と Google 画像検索の一致度は 35 %程度であると推測された．

6.2　モチーフの分類精度に関する結果

提案手法による画像分類精度を評価するため，学習済みの畳み込みニューラルネットワークを用いたモチーフの分類を行った．

モチーフの分類精度を Fig. 9 に示す．従来手法では学習の進行に関わらず 2 %程度の精度で収束した．仮に 50 カテゴリに対してランダムに分類を行った場合，理論上の分類精度は 100 / 50 = 2 %であるため，従来手法はランダムな分類と同等の結果となった．これに対し提案手法では学習が進むと精度が若干向上し，最終的に 15 %程度の精度が得られ，従来手法よりも分類精度の向上がみられた．

学習済みの畳み込みニューラルネットワークに対して各代表画像のモチーフと分類された小袖屏風分割画像の例を Fig. 10，Fig. 11，Fig. 12 に示す．それぞれ左の画像が代表画像，右の画像が代表画像のモチーフと分類された小袖屏風分割画像群である．Fig. 10 より，代表画像に対して画像特徴の類似した画像が同一モチーフであると分類された．具体的には，植物という点では桐と同じ物体が見られるが，モチーフの観点から見ると松や菊などのモチーフが桐と誤分類されている．Fig. 11，Fig. 12 も Fig. 10 と同様に画像特徴の類似した画像が分類されているが，異なるモチーフの画像を含んでいる．

6.3　考察

6.1 節，6.2 節において学習が進むと精度が収束したが，更なる精度向上が達成されなかった要因として，モチーフ間の画像特徴が類似していることが考えられる．提案手法では画像特徴空間における類似性のみを考慮して近傍空間を学習しているため，画像特徴が類似していてもモチーフが異なる場合には分類が困難となる．具体的には，小袖屏風中に含まれるモチーフは植物や山，滝といった自然物に偏りがあるため，画像特徴を考慮した近傍空間同士で重なりが発生しやすい傾向があると考えられる．さらに提案手法において単一のカテゴリのみを考慮して近傍空間の学習を行っている点についても，近傍空間の重なりが発生する要因として考えられる．6.2 節における分類画像例より，誤分類した画像でも画像特徴の観点から類似した画像が分類されていたため，画像特徴空間における近傍を考慮した学習となっている．本稿では画像特徴の類似性に対する評価が主観的であるため，さらに定量的な指標を加えることで提案手法に対する客観的評価が行える．

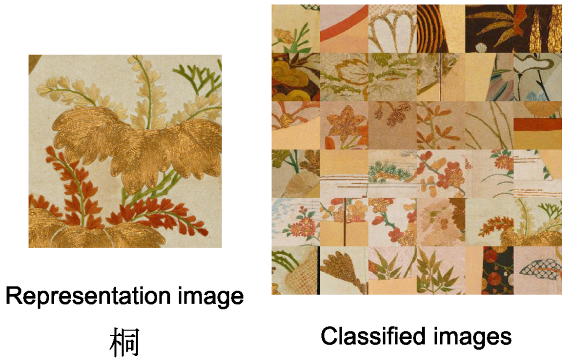


Fig. 10: Examples of motif classification 1.

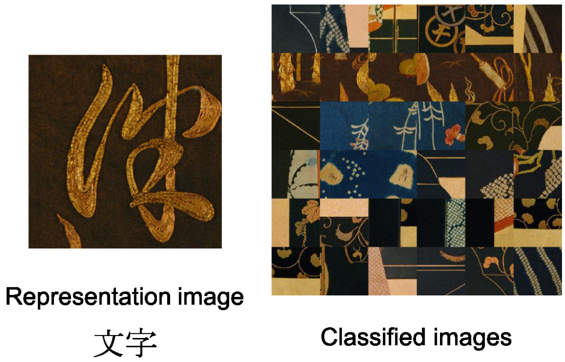


Fig. 11: Examples of motif classification 2.

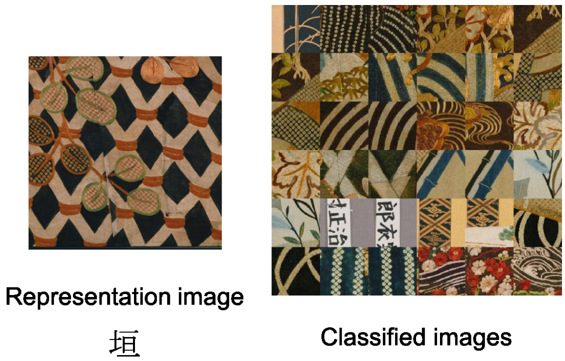


Fig. 12: Examples of motif classification 3.

7　おわりに

本稿では教師なし画像群に対して各分類カテゴリに属する代表画像を 1 枚ずつ選択し，Google 画像検索によって得られた画像群を用いて画像特徴空間上の近傍空間を学習する手法を提案した．実験により提案手法を用いて近傍空間を学習することによって，従来では困難であった教師なし画像群のカテゴリ分類が可能となることが示された．

しかし，画像特徴が類似しているがカテゴリが異なる場合には近傍空間が重なりあうため分類が困難であった．そのため近傍空間の重なりについては，各カテゴリの近傍空間の大きさを考慮した学習により重なりが緩和され，分類精度の向上が見込まれる．

参考文献

1) Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh： “A fast learning algorithm for deep belief nets.”, Neural Computation, 1527/1554 (2006)

2) Quoc V. Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S. Corrado, Jeff Dean, and Andrew Y. Ng： “Building high-level features using large scale unsupervised learning.” , International Conference on Machine Learning (2012)

3) Google 画像検索 , https://www.google.co.jp/imghp

4) Alex Krizhevsky and Sutskever, Ilya and Geoffrey E. Hinton： “ImageNet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems 25, 1097/1105 (2012)

5) 岡谷貴之： “機械学習プロフェッショナルシリーズ 深層学習”, 株式会社講談社 (2015)

6) 国立歴史民俗博物館： “国立歴史民俗博物館資料図録 2 野村コレクション 小袖屏風” (2002)

7) データベースれきはく , http://www.rekihaku.ac.jp/doc/t-db-index.html

AI帆走のためのパーティクルフィルタによるノイズ除去

○真鍋 秀朗，橘 完太（工学院大学）

Noise Reduction by Particle Filter for AI Sailing

* Hideaki Manabe and Kanta Tachibana (Kogakuin University)

Abstract－ The sailboat can be manoeuvered avoiding obstacle and arriving at destination if its position, direction, and absolute wind are measured exactly. However, measurement noise is added to signals when they are measured by GPS, compass, and wind sensors. We implement a sailing simulator to give observed values of position, direction, and apparent wind including the measurement noise to the AI sailor. This study investigates effect of particle filter to reduce the measurement noise.

Key Words: Noise Reduction, Particle Filter, AI Sailing

1 はじめに

今日，自動車や飛行機，電車などの自動運転技術が注目されているが，帆船の自動運転の研究はほとんどなされていない．自動車などの運転と帆船の運転の主な違いは，帆船は自らが動力を持たず，風や波の力を利用してしか動けないという点にある．また，自動車やモーターボートは位置，速度，加速度などが連続的に変化するが，帆船の場合は主動力となる風が不連続に変化することが頻繁に起こるため，通常の自動運転技術より複雑な問題である．

座標が既知の障害物を回避し，座標が既知の目的位置へ帆走するために帆船が取得するべき情報として，自艇の位置と向き，絶対風の方向が挙げられる．これらの情報が正確に分かれば，単純な制御ルールにより自動運転を実現できる．しかし，実機で用いる GPS，コンパス，風向風速センサーでは観測ノイズが生じる．そこで本研究では，自艇の位置と向き，見かけの風の観測値にノイズを付加するような帆走シミュレーターを実装し，パーティクルフィルタによるノイズ除去の効果を検証する．

著者の研究[1]では，帆走をするにあたり GPS，コンパス，風向センサーを考慮し，観測ノイズの大きさを変えてシミュレーション実験を行い，単純な制御ルールによる目的位置への帆走が不可能となる観測ノイズの閾値を検証した．しかし，この研究では実機で用いる観測ノイズの大体の大きさが判明したのみであり，その観測ノイズの低減に至らなかった．そこで本研究では，パーティクルフィルタ[2]を用いてノイズを除去し，不連続な状態遷移におけるノイズ除去の効果を検証する．

2 AI帆走と観測誤差

2.1 帆船操縦シミュレーター

本実験では，2D帆船操縦シミュレーターにより，AI帆走のためのパーティクルフィルタによるノイズ除去を目指す．Fig. 1は使用した帆船操縦シミュレーターのスクリーンショットである．

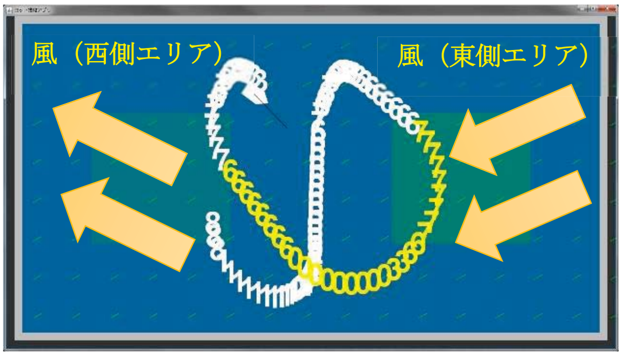


Fig. 1: 実験に使用した帆船操縦シミュレーター

帆船の真の加速度，速度，位置，方位は真の風のベクトル場とAIが操作する舵の影響を受ける．帆船の位置が変わると風ベクトルが変わり，さらに進行風も影響するため，帆船の状態は時々刻々変化する．

Fig. 2はシステムの概要図である．AIは帆船の位置$\vec{x}$，方位θ，風ベクトル$\vec{w}(\vec{x})$を観測する．その際，観測ノイズを考慮するために，真の情報にノイズを付加する．しかし，センサーには観測ノイズが付加されるため，AIに入力する前にパーティクルフィルタを通す．フィルタをかけた状態変数を用いてAIは次の行動を選択する．AIの取り得る行動は“左に舵を切る”，“右に舵を切る”，“舵を切らない（直進する）”の3種類である．行動によって帆船の方位が$\Delta\theta$だけ変化するため，次の時刻の帆船の位置や方位，風のベクトル場も変わることになる．$\vec{w}(\vec{x})$は，帆船の位置$\vec{x}$での絶対風$\vec{w}$を示す．これに対して見かけの風$\vec{a}$は，帆船が受ける風であり，絶対風から自艇の速度ベクトルを引いて求める($\vec{a}=\vec{w}-\vec{v}$)．

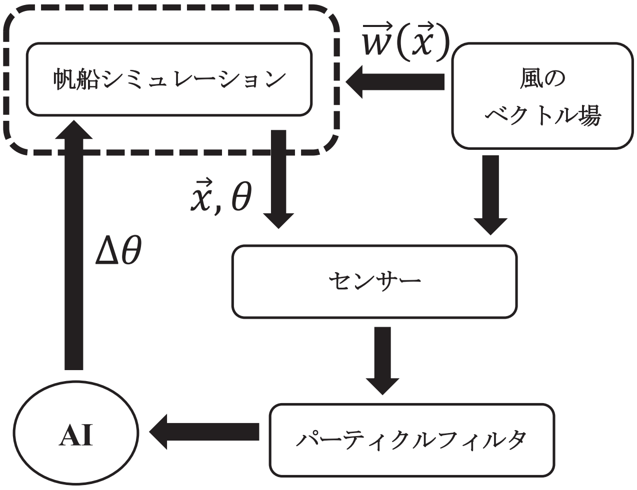


Fig. 2: システム概要図

Fig. 3に帆船のダイナミクスを示す．帆船の加速度は帆に受けうる推進力の大きさによって決まる．推進力はヨットが受ける見かけの風と帆船の方位から決まる．見かけの風は絶対風$\vec{w}$から自艇の速度ベクトル$\vec{v}$を引いたものであるから，$\vec{v}$と$\vec{w}$の値により，帆船の推進力が求まり，加速度$\dot{\vec{v}}$が決定する．$\dot{\vec{v}}$の値が変化すると$\vec{v}$の値も変化し，それにより次時刻の推進力が変わる．本研究の帆走シミュレーションでは，Fig. 2, Fig. 3の点線で囲んだモデルを実装する．

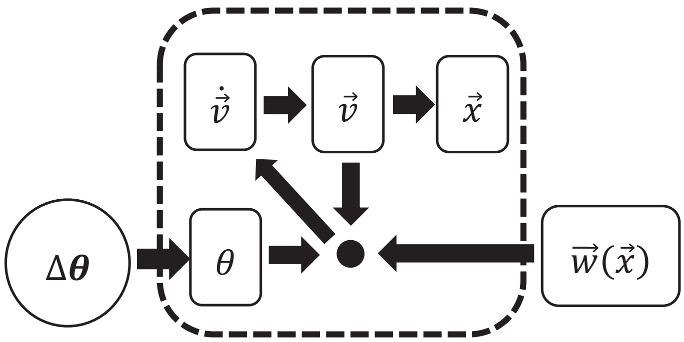


Fig. 3: 帆船のダイナミクス

2.2 センサーの観測ノイズ

GPSセンサー，風向風速センサー，方位センサーでそれぞれ$\vec{x}, \vec{a}(= \vec{w} - \vec{v}), \theta$を観測する．また，GPSセンサーの観測ノイズを$\vec{\epsilon}_x \sim N(\vec{0},\ {\sigma_x}^2 I)$，風向風速センサーの観測ノイズを$\vec{\epsilon}_a \sim N(\vec{0},\ {\sigma_a}^2 I)$，方位センサーの観測ノイズを$\epsilon_\theta \sim N(\beta, {\sigma_\theta}^2)$とする（$I$：2次元の単位行列）．

2.3 帆船の制御ルール

AIは単純なTable. 1に示す制御ルールにより，舵の操作$\Delta\theta$を決定する．

Table. 1: 帆船の制御ルール一覧

風向	状態	行動
右側	風下側前方に障害物がある	右
左側	風下側前方に障害物がある	左
右側	風上側前方に障害物がある	左
左側	風上側前方に障害物がある	右
右側	自艇の舳先がデッドゾーンを向いていて，かつ自艇から見た目的エリア全てがデッドゾーンに入っている	左
左側	自艇の舳先がデッドゾーンを向いていて，かつ自艇から見た目的エリア全てがデッドゾーンに入っている	右
-	前方に目的エリアの少なくとも一部がある	直進
-	右側に目的エリアの少なくとも一部がある	右
-	左側に目的エリアの少なくとも一部がある	左
-	上記のいずれでもない	直進

風向は自艇の舳先に向いて見た風の方向を指す．自艇の舳先に向いて右側から風を受ける場合は“右側”となり，左側から風を受ける場合は“左側”となる．舵の操作$\Delta\theta$において，“右”の場合は$\Delta\theta = 1$とし，“左”の場合は$\Delta\theta = -1$とする．“直進”の場合は$\Delta\theta = 0$となる．なお，“前方”は帆船の舳先の左右$\pm 60°$の範囲，“デッドゾーン”は帆船から見た絶対風の方向$\pm 45°$の範囲とする．

3　パーティクルフィルタ

パーティクルフィルタはリサンプリング→予測→観測→重み付け→リサンプリング…の順に行われる．

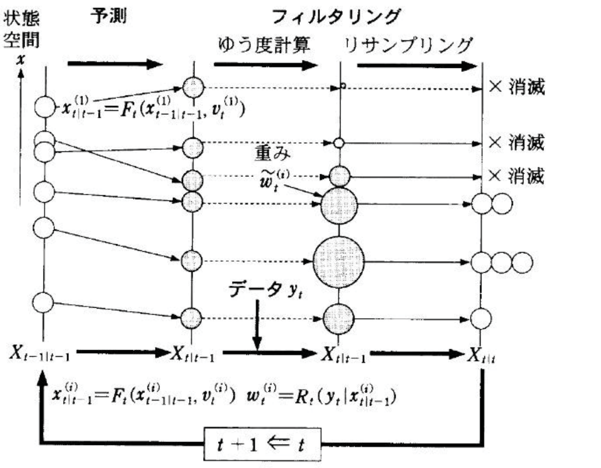


Fig. 4: パーティクルフィルタ
アルゴリズムの概略図（樋口(2005)[2)]）

多数のパーティクルから次時刻のパーティクルを予測し，観測値と照らし合わせて尤度を計算する．そしてリサンプリングにより，尤度の高いパーティクルが高い確率で選ばれ，尤度の低いパーティクルはあまり選ばれなくなる．Fig. 4 はパーティクルフィルタアルゴリズムの概略図である．本章では本研究で使用したシステムモデルと観測モデルについて説明する．

3.1 システムモデル

状態変数として，帆船の位置$\vec{x}_t$，速度$\vec{v}_t$，方位θ_t，風ベクトル$\vec{w}_t$を使用する．舵の操作量に応じて帆船が等速運動し，絶対風が常に一定であるとする以下のシステムモデルを用いる．

$$\vec{v}_t = \vec{v}_{t-1} + \vec{\varepsilon}_v^{sys}$$

$$\vec{x}_t = \vec{x}_{t-1} + \vec{v}_t + \vec{\varepsilon}_x^{sys}$$

$$w_t = \vec{w}_{t-1} + \vec{\varepsilon}_w^{sys}$$

$$\theta_t = \Delta\theta_t + \arctan\frac{\sin\theta_{t-1} + \varepsilon_{\theta,2}^{sys}}{\cos\theta_{t-1} + \varepsilon_{\theta,1}^{sys}}$$

$$\vec{\varepsilon}_x^{sys} \sim N\left(\vec{0}, {\sigma_x^{sys}}^2 I\right)$$

$$\vec{\varepsilon}_v^{sys} \sim N\left(\vec{0}, {\sigma_v^{sys}}^2 I\right)$$

$$\vec{\varepsilon}_w^{sys} \sim N\left(\vec{0}, {\sigma_w^{sys}}^2 I\right)$$

$$\varepsilon_{\theta,1}^{sys}, \varepsilon_{\theta,2}^{sys} \sim N\left(0, {\sigma_\theta^{sys}}^2\right)$$

時刻tにおける舵の操作$\Delta\theta_t$は，ノイズ無しで観測できるものとする．

3.2 観測モデル

観測変数として，帆船の位置$\vec{x}_t^{obs}$，見かけの風ベクトル$\vec{a}_t^{obs}$，方位θ_t^{obs}を使用する．以下の観測モデルを用いる．

$$\vec{x}_t^{obs} = \vec{x}_t + \vec{\varepsilon}_x^{obs}$$
$$\vec{a}_t^{obs} = R(-\theta_t)(\vec{w}_t - \vec{v}_t) + \vec{\varepsilon}_a^{obs}$$
$$\theta_t^{obs} = \Delta\theta_t + \arctan\frac{\sin\theta_t + \varepsilon_{\theta,2}^{obs}}{\cos\theta_t + \varepsilon_{\theta,1}^{obs}}$$

$$\vec{\varepsilon}_x^{obs} \sim N\left(\vec{0}, {\sigma_x^{obs}}^2 I\right)$$

$$\vec{\varepsilon}_a^{obs} \sim N\left(\vec{0}, {\sigma_a^{obs}}^2 I\right)$$

$$\varepsilon_{\theta,1}^{obs}, \varepsilon_{\theta,2}^{obs} \sim N\left(0, {\sigma_\theta^{obs}}^2\right)$$

（ただし，$R(\theta)$：回転行列）

3.3 尤度計算

観測モデルに基づき，各パーティクルの尤度Lを計算する．尤度Lは以下の式で表される．

$$L = \exp(-|\vec{z}_x|^2 - |\vec{z}_a|^2 + \beta\cos(\theta_t^{obs} - \theta_t))$$

ここで，$\vec{z}_x, \vec{z}_a$は以下である．

$$\vec{z}_x = \frac{\vec{x}_t^{obs} - \vec{x}_t}{\sigma_x^{obs}}$$

$$\vec{z}_a = \frac{\vec{a}_t^{obs} - R(-\theta)(\vec{w}_t - \vec{v}_t)}{\sigma_a^{obs}}$$

なお，$\beta\cos(\theta^{obs} - \theta)$はフォン・ミーゼス分布の確率密度関数

$$f(\theta^{obs}) = \frac{\exp\{\beta\cos(\theta^{obs} - \theta)\}}{2\pi I_0(\beta)}$$

において，定数$\frac{1}{2\pi I_0(\beta)}$を削除したものである．$\beta$は集中度パラメータである．

3.4 リサンプリング

リサンプリングはルーレット選択により行う．尤度の高いパーティクルは選ばれやすく，尤度の低いパーティクルは選ばれにくくなる．

4　実験

本実験では，1,600×900 ピクセルのフィールドの中の東側と西側にある 2 つの目的エリアを交互に周回させ，帆船に搭載したセンサーに付加されるノイズを低減することを目標とする．片方の目的エリアの中に帆船が完全に入ると，反対側のエリアが目的エリアに変わり，それが繰り返される．なお，不連続な状態遷移に対するパーティクルフィルタの効果を検証するため，フィールドの西半分の風向を東南東，東半分の風向を東北東とし，風速は一定とする．

観測ノイズのパラメータは，$\sigma_x = 10$，$\sigma_w = 0.3$，$\sigma_\theta = 3$と設定した．また，パーティクルフィルタのパーティクル数は 10,000 とする．システムモデルにおける$\sigma_x^{sys}, \sigma_v^{sys}, \sigma_w^{sys}$の 3 つのパラメータのうち 1 つの値のみを変動させて実験を行う．また，システムモデルにおいて，$\sigma_\theta^{sys} = 10$，観測モデルにおいて$\sigma_x^{obs} = 10, \sigma_a^{obs} = 0.3, \beta = 10$に設定する．

帆船の真の位置を$\hat{\vec{x}}$，速度を$\hat{\vec{v}}$，絶対風を$\hat{\vec{w}}$とし，センサーを通して観測した値をそれぞれ$\vec{x}, \vec{v}, \vec{w}$，各パーティクルから算出された予測値をそれぞれ$\bar{\vec{x}}, \bar{\vec{v}}, \bar{\vec{w}}$とする．実験は各パラメータで 10,000 ステップ行い，最後の 1,000 ステップの観測値と真の値の差の二乗平均平方根

$$\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{x}_t - \hat{\vec{x}}_t\right|^2},\ \sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{v}_t - \hat{\vec{v}}_t\right|^2},$$

$$\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{w}_t - \hat{\vec{w}}_t\right|^2}$$

，予測値と真の値の差の二乗平均平方根

$$\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{x}}_t - \hat{\vec{x}}_t\right|^2},\ \sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{v}}_t - \hat{\vec{v}}_t\right|^2},$$

$$\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{w}}_t - \hat{\vec{w}}_t\right|^2}$$

ノイズ残存率

$$\frac{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{x}_t - \hat{\vec{x}}_t\right|^2}}{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{x}}_t - \hat{\vec{x}}\right|^2}},\ \frac{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{v}_t - \hat{\vec{v}}_t\right|^2}}{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{v}}_t - \hat{\vec{v}}_t\right|^2}},$$

$$\frac{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\vec{w}_t - \hat{\vec{w}}_t\right|^2}}{\sqrt{\frac{1}{1000}\sum_{t=9001}^{10000}\left|\bar{\vec{w}}_t - \hat{\vec{w}}_t\right|^2}}$$

を各実験で算出する．ノイズ残存率とは，それぞれの状態変数について観測値と真の値の差の二乗平均平方根を予測値と真の値の差の二乗平均平方根で割った値であり，観測値と比べたパーティクルフィルタによるノイズの残り度合を示した値である．

5　結果

Fig. 5 - Fig. 7 に，パラメータをそれぞれ$\left(\sigma_x^{sys}, \sigma_v^{sys}, \sigma_w^{sys}\right)$と設定した際の結果を示す．Fig. 5 に，$\sigma_v^{sys} = 0.5, \sigma_w^{sys} = 1$として，$\sigma_x^{sys}$を変動させて行った各実験の最後の 1,000 ステップの位置についてノイズ残存率を示す．Fig. 6 に，$\sigma_x^{sys} = 5, \sigma_w^{sys} = 1$として，$\sigma_v^{sys}$を変動させて行った各実験の速度についてノイズ残存率を示す．Fig. 7 に，$\sigma_x^{sys} = 5, \sigma_v^{sys} = 0.5$として，$\sigma_w^{sys}$を変動させて行った各実験の最後の 1,000 ステップの風ベクトルについてノイズ残存率を示す．ノイズ残存率は実験終了までの 1,000 ステップの平均を折れ線で表す．

また，各ステップのノイズ残存率

$$\frac{\left|\vec{x}_t - \hat{\vec{x}}_t\right|}{\left|\bar{\vec{x}}_t - \hat{\vec{x}}\right|},\ \frac{\left|\vec{v}_t - \hat{\vec{v}}_t\right|}{\left|\bar{\vec{v}}_t - \hat{\vec{v}}_t\right|},\ \frac{\left|\vec{w}_t - \hat{\vec{w}}_t\right|}{\left|\bar{\vec{w}}_t - \hat{\vec{w}}_t\right|}$$

を実験終了までの 1,000 ステップ分求めた値の第 1 四分位と第 3 四分位をエラーバーで示す．

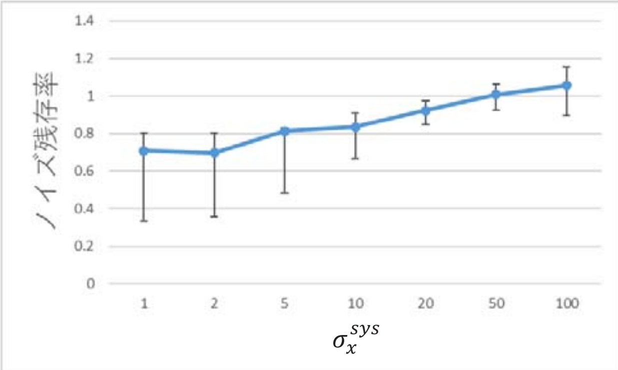


Fig. 5: パラメータを$(\sigma_x^{sys},\ 0.5,\ 1)$とした際の位置のノイズ残存率

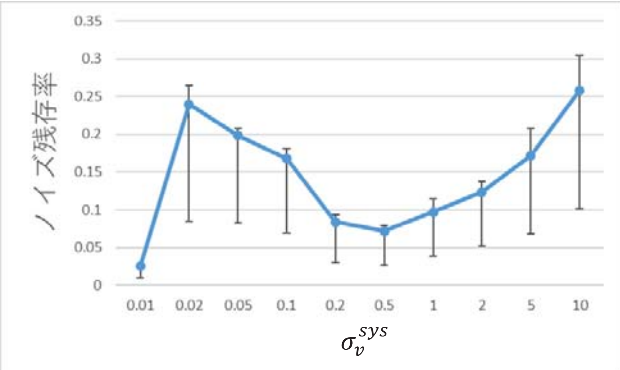


Fig. 6: パラメータを$(5,\ \sigma_v^{sys},\ 1)$とした際の速度のノイズ残存率

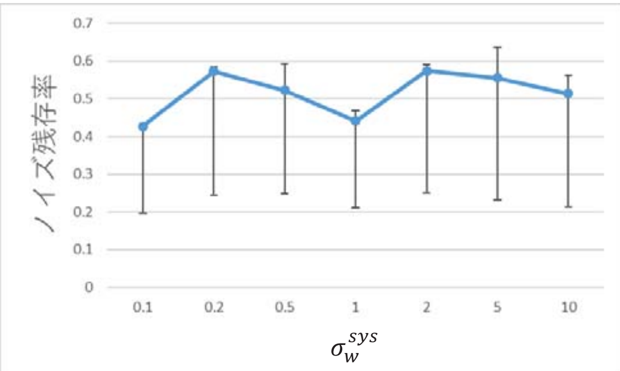


Fig. 7: パラメータを$(5,\ 0.5,\ \sigma_w^{sys})$とした際の風向風速のノイズ残存率

6　考察

本実験では，不連続な状態遷移におけるノイズ除去の効果を検証するため，風ベクトルの境目における帆船の動きと，その際に真の風ベクトル，観測値の風ベクトル，パーティクルフィルタをかけてノイズを除去した風ベクトルの3種類について考察する．

Fig. 8 にパラメータを$(5,\ 0.5,\ 1)$とした際の実験において，10,000ステップに到達する直前の東西のフィールドの境目を通過した直前3ステップと直後3ステップにおける風ベクトル$\vec{w}$を示す．横軸がx方向（東西），縦軸がy方向（南北）である．黒色の点はステップごとの帆船の位置，青色の直線は真の$\vec{w}$，橙色の直線は観測値からそのまま算出した$\vec{w}$，灰色の直線はパーティクルフィルタをかけてノイズを除去した$\vec{w}$である．ノイズをそのまま反映させた$\vec{w}$では，観測ノイズの影響で真の風ベクトルから大きな隔たりがあるが，パーティクルフィルタをかけた$\vec{w}$では，真の値に近づくことが分かる．よって，状態空間が不連続の場合でも良好にノイズ除去ができたといえる．

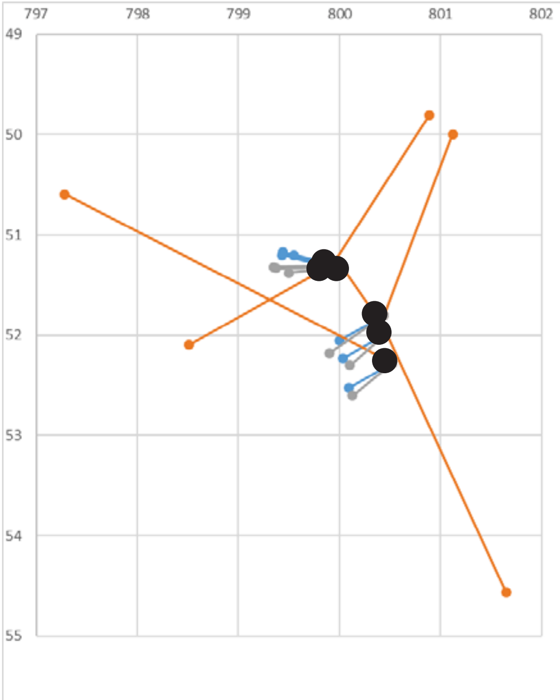


Fig. 8: フィールドの境目を通過した際の$\vec{w}$

7　まとめ・今後の課題

本研究では，状態遷移が不連続に変化することが頻繁に起こる帆船のAI帆走という複雑な問題を考えるため，パーティクルフィルタによるノイズ除去を実現した．また，それぞれのパラメータを1つだけ変えた際のノイズ残存率を計測した．

今回はパラメータのチューニングを手動で行ったが，今後はこれらのパラメータのチューニングを自動的に行い，センサーに最も適したパーティクルフィルタを実装したいと考えている．

参考文献

1) 真鍋 秀朗, 橘 完太：AI帆走に必要なセンサー性能の検証, 計測自動制御学会 システム・情報部門 学術講演会(SSI2015), SS21-13 (2015)

2) 樋口 知之：粒子フィルタ, 電子情報通信学会誌, **Vol.88**, No.12, pp.989-994 (2005)

軍隊アリシステムの利他行動を取り入れたソーシャルコミュニティの成長促進に対する提案 -オープンソースソフトウェアにおけるグループ開発手法への適用-

○上本拓也 (県立広島大学大学院総合学術研究科経営情報学専攻)
市村匠 (県立広島大学経営情報学部経営情報学科)

A Proposal for Growing Process of Social Community by Altruism Behavior in Army Ant System -Its Application to Group Development Method for Open Source Software-

*T. Uemoto (Graduate School of Comprehensive Scientific Systems, Prefecture University of Hiroshima)
T. Ichimura (Department of Management and Information Systems, Prefecture University of Hiroshima)

Abstract− Recently, the developing software project is complex tasks because many projects in developing process are processed parallel and many developers join two or more projects. To improve the efficiency of group developing software, various methods are executed and the analysis of their behaviors is important. The community of the developers are analyzed by using analyzing method of network structure. We develop the simulation system of developing software with altruism behaviors. The system was developed on the basis of developers behaviors of SourceForge.net. In this paper, we observed the community growing process under the agent's altruism behaviors. This paper reports the efficiency of altruism behavior in the software developing community.

Key Words: Altruism behaivor, Social community, Software development

1　はじめに

近時，ソフトウェア開発において，複数の開発者が１つのグループとなりコーディングが行われるチーム開発が主流である．チームは１つ以上のプロジェクトを実施し，それらを統合することで規模の大きなソフトウェアを構築する．企業におけるソフトウェア開発ではなく，オープンソースソフトウェアにおいては，グループをコミュニティとし，開発者がコミュニティに自身で参加する形式がとられている．この過程を複雑ネットワークにおけるコミュニティの成長過程ととらえ，分析した研究が Madey らによって行われている [1)]．文献 [1)] では，シミュレーションシステムを開発するにあたり，SourceForge.net の開発者やプロジェクトに関するオープンデータ [2)] を用いた．オープンソースソフトウェア開発におけるシミュレーションシステムでは，エージェントは開発者を示し，これらがソフトウェア開発のためにプロジェクトを形成する．エージェントの行動には，'プロジェクトを生成する'，'プロジェクトに参加する'，'プロジェクトを辞退する'，'行動は起こさず現状の開発に取り組む' の４種類がある．エージェントはプロジェクトに参加して他のエージェントと共に開発に携わるころで，１つのコミュニティを形成し，それらが複数集まることでソーシャルコミュニティを形成する．このようなシミュレーションを用いて，Madey らはオープンソースソフトウェア開発におけるソーシャルコミュニティが形成されていく過程を観察した．

本研究では，まず，エージェントの行動の種類を取り入れたシミュレーションシステムを開発した．ここでは，システムにソフトウェアのタスク量やエージェントのスキルを取り入れた．このシステムを用いて，形成されたコミュニティとエージェントの行動の関連性を調査した [3)4)]．タスクとはプログラミングなどソフトウェア開発における仕事の量を表したものであり，スキルとは単位時間あたりに仕事をどの程度処理できるかという能力を表したものである．シミュレーション結果 [3)4)] より，ソフトウェア開発に参加するエージェント数が多いほどソフトウェアの数が減少する分布が得られ，極座標方程式を用いて分布の近似とその拡がりを求めた．また，'プロジェクトを辞退する' といったエージェントの行動とソフトウェアのタスク量の関連性について分析を行った．タスク量が少ないソフトウェアでは辞退するエージェント数が少なく，より多くのソフトウェアにおいては多くのエージェントが辞退し，ある量のタスクを越えたソフトウェアにおいては辞退するエージェント数が少ないことがわかった．多くのエージェントが辞退するソフトウェアにおいて，辞退する数を抑え，全体としてより開発に適したコミュニティの形成手法の必要性が議論されていた．

本論文では，ソフトウェア開発におけるコミュニティの成長を促進するためにエージェントの行動に利他行動 [7)] を実現するための機能を追加する．利他行動とは，仲間の利益のために自己を犠牲にする行動である．能力が高く，時間に余裕があるエージェントは，タスク量は多いがエージェント数が少ないと思われるプロジェクトを発見すると，利他行動として開発に参加する．自

身の負担は増えるが他のエージェントに協力することで，他のエージェントの負担の減少にともないプロジェクトを辞退する数が減少し，全体として開発効率が向上するのではないかと考えられる．利他行動を取り入れたシミュレーションシステム [5] を用いて，辞退するエージェント数や全体の開発時間などの点で調査を行った．結果として，形成されたコミュニティにおけるタスク量とエージェントの行動の関連性の抽出と利他行動によるコミュニティの成長と開発時間の短縮が見られたため，本論文にて報告する．

2 SourceForge.net

SourceForge.net はオープンソースソフトウェアのソースレポジトリであり，ソフトウェアのダウンロードやソフトウェア開発の管理システムの利用が可能である．ソフトウェア開発を行うプロジェクトや開発者などの情報はデータベース [6] に保管されている．2014 年 9 月時点でデータベースには 'user_group' テーブルを含む 73 個のテーブルがある．'user_group' テーブルとは，開発者固有の ID である 'user_id' やプロジェクト固有の ID である 'group_id' などが記録されており，開発者とプロジェクトの関連性に関するデータが記録されている．

本研究では 'user_group' テーブルより，'group_id' をもとにすべてのレコードをカウントすることで，プロジェクトに参加している開発者数や，開発者数に関するプロジェクトの数について調査を行った．2004 年 9 月時の調査結果を Fig. 1 に示す．横軸はプロジェクトに参加している開発者数，縦軸はその開発者数におけるプロジェクト数であり，それぞれの値の対数をとった値を示している．Fig. 1 によると，ある点までは線形的に減少し，それ以降は 2 つの直線で表される幅を持った分布となった．この調査結果はシミュレーションシステムにおけるパラメータなどを定義するために 3 節で使用される．

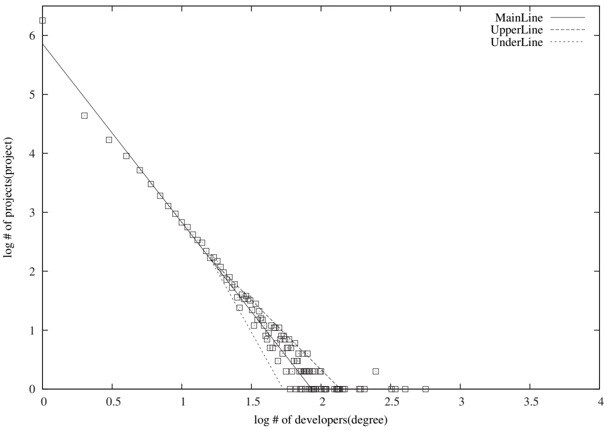


Fig. 1: SourceForge.net における，プロジェクトに参加している開発者数とプロジェクト数

3 シミュレーションシステムの概要

本研究で提案するシミュレーションシステムでは，メジャーエージェントとマイナーエージェントを定義する．メジャーエージェントはプロジェクトの生成を行い，マイナーエージェントがプロジェクトに参加・辞退を行うことでソーシャルコミュニティを構築する．メジャーエージェントはプロジェクトマネージャ，マイナーエージェントは開発者を表す．メジャーエージェントは開発するソフトウェアの設計を行うとタスク量を決定する．マイナーエージェントはチームとしてそのタスクを処理する．

SourceForge.net はソフトウェアを 10 個のカテゴリで分類している [2]．シミュレーションシステムにおいても同様にソフトウェアのカテゴリを定義し，メジャーエージェントはソフトウェアを生成する際，10 個のカテゴリの中から 1 つのカテゴリを選択する．ソフトウェア開発において複数のタスクがあり，カテゴリ特有の 1 種類のタスクと全カテゴリに共通する 3 種類のタスクとする．カテゴリ特有のタスクとは専門的な知識を要するタスクのことである．全カテゴリ共通のタスクとは，ネットワーク・通信処理，データベース処理，グラフィック処理であり，特定のカテゴリに限らず全てのソフトウェア開発に共通するタスクである．

マイナーエージェントはタスクに関するスキルを持ち，スキルとは単位時間あたりにどの程度のタスクを処理できるかを表す．ソフトウェアのカテゴリ特有のタスクを処理するためのスキルに関して，マイナーエージェントは 10 種類のうち複数の種類のスキルを持っており，知識を持っていないカテゴリに関するスキルは 0 である．全てのマイナーエージェントは全カテゴリに共通するタスクを処理するためのスキルを持っている．マイナーエージェントがプロジェクトに参加する際，そのプロジェクトで開発を行うソフトウェアのカテゴリのスキルを持たないエージェントは参加することができない．1 つのプロジェクトに複数のエージェントが参加している場合，タスクは均等に与えられる．

3.1 タスクとスキル

プロジェクトのタスク量とエージェントのスキルを決定するパラメータを定義する．プロジェクト i に与えられるタスクのパラメータは P_i^k であり，エージェント j に与えられるスキルのパラメータは S_j^k である．

3.1.1 タスクに関するパラメータ

プロジェクトにはソフトウェア開発中の 1 種類以上の開発タスクを割り当てられる．P_i^k はプロジェクト i の開発タスク k のタスク量であり，$[0.0, 2.0]$ で与えられる．プロジェクト i のタスク量 $\mathbf{P_i}$ はベクトルであり，要素は割り当てられた開発タスクのタスク量 P_i^k を用いて $\mathbf{P}_i = \{P_i^1, P_i^2, ..., P_i^n\}$ と表される．プロジェクトに複数のエージェントが参加している場合，エージェント 1 個あたりのタスク量は以下のように求められる．

$$W_i^k = \frac{P_i^k}{N_i} \tag{1}$$

ここで，W_i^k はプロジェクト i に参加したエージェントに割り当てられるタスク k に関するタスク量，P_i^k はプロジェクト i のタスク k のタスク量，N_i はプロジェクト i に参加しているエージェントの数である．

3.1.2 スキルに関するパラメータ

S_j^k はエージェントの開発タスクごとの単位時間あたりに処理が可能なタスク量であり，初期値は $[0.0, 1.0]$ でランダムに与えられる．エージェントに割り当てら

れたタスク量を S_j^k で割ることで開発に関わる時間が求められる．参加しているプロジェクトごとに開発時間を求め，その和 T_j はエージェント j が開発に費やす時間の合計を表す．

$$T_j = \sum_i T_{ij} = \sum_k \frac{W_{ij}^k}{S_j^k} \tag{2}$$

ここで，W_{ij}^k はエージェント j に割り当てられたプロジェクト i の開発タスク k のタスク量である．1 ステップにおける時間は有限であり，T_j はパラメータ T_{limit} を用いて $0 \leq T_j \leq T_{limit}(0 < T_{limit} \leq 24)$ という制約が与えられる．T_j が T_{limit} を越えてしまう場合はエージェントは新しくプロジェクトに参加することはできない．

4 エージェントの行動

文献[1]において，エージェントの行動は 4 種類ある．本研究で提案するシミュレーションモデルではメジャーエージェントとマイナーエージェントを定義し，4 種類の行動を 2 つのエージェントに割り当てる．

- メジャーエージェント
  - プロジェクトを生成する
- マイナーエージェント
  - プロジェクトに参加する
  - プロジェクトを辞退する
  - 何も行動を起こさない

マイナーエージェントは 1 ステップで 3 つの行動から 1 つを選択し，それぞれの行動アルゴリズムに従って行動する．

4.1 プロジェクトを生成する

メジャーエージェントがプロジェクトを生成する状況は 2 つある．

- メジャーエージェント j が生成したプロジェクトにおいて，エージェントの負担が大きい場合，そのプロジェクトのタスクを担うサブプロジェクトを生成する．
- 新しいソフトウェアが設計され，開発するためのプロジェクトを生成する．

新しいプロジェクトを生成する場合，ランダム値を発生させ，確率 $P(New)$ と比較し，新しいソフトウェアを開発するためのプロジェクトを生成する．サブプロジェクトを生成する場合，評価式 Sub を求め，しきい値と比較を行うことでサブプロジェクトを生成するかどうかを決定する．メジャーエージェントはサブプロジェクトを生成する時，もととなるプロジェクトのエージェント 1 個あたりの負荷を計算し，しきい値と比較してサブプロジェクトを生成するかどうかを決定する．負荷の計算は各タスクごとに計算を行う．

$$Sub_i^k = \frac{P_i^k}{N_i^k} \tag{3}$$

$$\begin{cases} IF \ \ Sub_i^k \geq Sub_{threshold} \ \ THENCreateProject \\ IF \ \ Sub_i^k < Sub_{threshold} \ \ THENNotCreate \end{cases}$$

Sub_i^k は，プロジェクト i の開発タスク k におけるエージェント 1 個あたりのタスク量であり，P_i^k はプロジェクト i の開発タスク k に関して割り当てられたタスク量，N_i^k はプロジェクトに参加しているエージェント数，$Sub_{threshold}$ はタスクを分散するためのしきい値である．各開発タスクに関して式 (3) により判別を行い，プロジェクトのどのタスクを分散するかを決定する．決定した分散するタスクを持つサブプロジェクトを生成する．

4.2 プロジェクトに参加する

エージェント j は，自分が参加していないプロジェクトを 1 つ選択する．選択されたプロジェクトに対して参加する確率を求め，しきい値と比較することで参加することを決定する．参加することを決定した場合でも，開発に費やす時間 T_j を再計算し，T_j が制約である T_{limit} を越える場合，プロジェクトに参加することをやめる．

4.2.1 プロジェクトの選択方法

本研究では，エージェントの行動にネットワーク成長モデルである動的適応度を持つ Barabasi-Albert(BA) モデル[8]を適用する．動的適応度を持つ BA モデルとは，次数が高い点に高い確率で辺が繋がれる優先的選択を取り入れた BA モデル[9]に動的適応度を加えたものである．ソフトウェアに参加するエージェントをソフトウェアの次数とし，エージェントが参加するソフトウェアを決定する際にこのモデルを適用する．ソフトウェア j を選択する確率は以下の通りである．

$$S_j = \frac{\eta_j N_j}{\sum \eta_j N_j} \tag{4}$$

ここで，η_j はソフトウェアの適応度，N_j はソフトウェアに参加しているエージェント数である．適応度 η_j は時間経過とともに減少する値であり，適応度を動的に変化させることで，コミュニティの成長・安定・減退を表すことができる．

4.2.2 プロジェクトに参加する確率

プロジェクトに参加する確率は，プロジェクトのタスク量とエージェントが持つスキルよりタスクの達成度が求められる．この時，プロジェクトのカテゴリに関するエージェントのスキルが 0 の場合は確率を 0 とし，スキルを持っている場合はその値に基づいて計算を行う．プロジェクト i に参加する確率は以下の通りである．

$$J_{ij} = \frac{\sum_{k=1}^{n}(P_i^k - \sum_{j=1} S_j^k)}{\sum_{k=1}^{n} P_i^k} \tag{5}$$

$$\begin{cases} IF \ \ J_{ij} \geq J_{threshold} \ \ THENJointProject \\ IF \ \ J_{ij} < J_{threshold} \ \ THENNotJoin \end{cases}$$

$\sum_{k=1}^{n}(P_i^k - \sum_{j=1} S_j^k)$ は，プロジェクト j のタスク量からエージェントのスキルを引くことで，その時点でどの程度のタスクが未達成であるかを求め，全体のタスク量 $\sum_{k=1}^{n} P_i^k$ で割ることで割合を求める．この値がしきい値 $J_{threshold}$ を越えるとエージェントはプロジェクトに参加する．

4.3 プロジェクトを辞退する

マイナーエージェント j がプロジェクトに参加する行動のアルゴリズムは以下の通りである．

1. エージェント j は，自分が参加しているプロジェクトを 1 つ選択する．
2. 選択されたプロジェクトに対して辞退する確率を求める．
3. プロジェクトを辞退するしきい値と比較し，辞退するかどうかを決定する．

手順 2 で求められる確率は以下の式 (6) で求められる．

$$L_{ij} = \sum_{k=1}^{n} \left(\frac{P_i^k}{N_i^k} - S_j^k\right) \quad (6)$$

$$\begin{cases} IF \ \ L_{ij} \geq L_{threshold} \ \ THEN LeaveProject \\ IF \ \ L_{ij} < L_{threshold} \ \ THEN NotLeave \end{cases}$$

$\frac{P_i^k}{N_i^k}$ はタスク k に関してエージェント 1 個あたりに割り当てられたタスク量であり，エージェント j のスキル S_j^k との差を求めることでエージェントの負荷の大きさを求める．$L_{threshold}$ はしきい値であり，エージェントの負荷 (L_{ij}) がこの値を越えるとプロジェクトを辞退する．

5 利他行動によるプロジェクトへの参加

利他行動とは仲間の利益のために自己を犠牲にする行動である．ソフトウェア開発のシミュレーションシステムの利他行動では次のような挙動が見られる．マイナーエージェントは自身の負担は増え参加しているプロジェクト自体の負荷が増えることになる．しかし，タスク量に対して参加しているエージェント数が少ない他のプロジェクトに参加する．つまり，エージェント j_1 が参加しているプロジェクト i_1 のエージェントが，エージェント j_2 が参加しているプロジェクト i_2 に協調的に参加する．これにより個々のソフトウェアの開発効率は低下するものの，全体の開発効率の向上につながると考えられる．

5.1 エージェント間の関連性

シミュレーションを通して形成されるコミュニティにおいて，エージェントはプロジェクトに参加するとき，プロジェクトにすでに参加している他のエージェント 1 個を選択し，リンクを張る．リンクがエージェント間の関連性を表しており，2 個のエージェント間にリンクが直接的に張られていなくても，いくつかのエージェントを介することで希薄ではあるが関連性が見られる．このようにエージェント間に存在するリンクの数を距離と定義する．

5.2 行動アルゴリズム

マイナーエージェント j_1 が利他行動を行うアルゴリズムは以下の通りである．

1. ランダム値を発生させ，確率 $P(Alt)$ と比較し，利他行動を行うかどうかを決定する．
2. 利他行動を行うことを決定した場合，距離 r 以内でリンクしているエージェント j_2 を選択する．
3. エージェント j_1 が参加しているプロジェクト i_1 の全てのエージェントは，選択されたエージェントが参加しているプロジェクト i_2 に関して評価式を求め，プロジェクト i_2 に参加するかどうかを決定する．

ここで，$P(Alt)$ および r はあらかじめ与えられる値である．手順 3 において用いられる評価式は以下の通りである．

$$R_{lm} \times \frac{P_i}{N_i} \geq A_{threshold} \quad (7)$$

ここで，R_{lm} はプロジェクト i_1 のエージェントがプロジェクト i_2 のエージェントと r 以内の距離でリンクがあれば 1，そうでなければ 0 である．P_i はプロジェクト i_2 のタスク量，N_i はプロジェクト i_2 のエージェント数，$A_{threshold}$ はしきい値である．利他行動で参加する場合も，T_j が T_{limit} を越える場合は参加できない．

6 シミュレーション

ソフトウェアのタスクやエージェントのスキルを取り入れ，シミュレーションを通して形成されたコミュニティとエージェントの行動の関連性の調査を行う．メジャーエージェントは 1,000 個，マイナーエージェントは 20,000 個で行う．マイナーエージェントの行動アルゴリズムに用いられるパラメータにおいて，$J_{threshold} = \{0.0, 0.5, 1.0\}$，$L_{threshold} = \{0.0, 0.5, 1.0\}$ を適用した 9 パターンのパラメータセットでシミュレーションを行った．

6.1 シミュレーション結果

シミュレーションによって形成されたコミュニティに関して，参加しているエージェント数とソフトウェア数の関連性を Fig. 2(a) に示す．横軸は参加しているエージェントの数の対数，縦軸はソフトウェアの数の対数である．Fig. 2(a) にはシミュレーション結果に加え，分布を示す曲線が描かれており，9 パターンのパラメータセットにおいてこのような近似曲線が描かれることがわかった．

近似曲線は，極座標方程式によって求められる曲線である．また，横軸のある点を境に幅 δ を持った分布を示している．極座標方程式および幅 δ は角度 θ を用いて以下の式で求められる．

$$\begin{aligned} r &= a(\theta) \times \theta & (8) \\ a(\theta) &= 0.07 - 0.07 \times \frac{\theta}{2\pi} & \\ r &= a(\theta) \times \theta - \delta & (9) \\ \delta &= 1.6 \times \frac{\theta - \theta_1}{2\pi} \ (\theta_1 \leq \theta) & \end{aligned}$$

ここで，θ_1 とは分岐を開始する点を表す角度を表す．どのようなパラメータにおける結果でも，式 (8) と式 (9) で表される曲線で近似されることが分かった．

次に，ソフトウェアに参加しているエージェント数とマイナーエージェントの行動の関連性について考察

する．Fig. 2(b) はソフトウェアのエージェント数とソフトウェアに参加したエージェント数と辞退したエージェント数の関連性を同時に表している．横軸はソフトウェアのエージェント数，縦軸はそれぞれの行動したエージェント数を表す．参加と辞退，どちらにおいてもエージェント数が多いソフトウェアほど値が線形的に上昇している．また，ソフトウェアのエージェント数とソフトウェアのタスク量およびソフトウェアの適応度の関連性について Fig. 2(c) に表す．横軸はソフトウェアのエージェント数，縦軸はそれぞれの値を表す．タスク量と適応度が高いほどソフトウェアのエージェント数が高くなることを表す．式 (4) より，適応度が高いほどエージェントに選択されやすくなり，かつ，式 (5) よりタスク量が多いほど参加しやすくなるためだと考えられる．

Fig. 2(d) はソフトウェアのタスク量と辞退したエージェントの関連性を示している．横軸はソフトウェアのタスク量，縦軸は辞退したエージェントの数である．タスク量が $[0.0, 1.5]$ の範囲では，ソフトウェアを辞退したエージェントは少ない．これは，エージェント 1 個あたりのタスク量が小さいため，開発に参加していても負荷が少ないと判断するためである．タスク量が $[1.5, 5.0]$ の範囲では，多くのエージェントが辞退している．これはエージェント 1 個あたりのタスク量が大きく，辞退するエージェントが増えたためである．式 (6) より，タスク量が多い場合だけでなく，エージェントのスキルが小さい場合に確率が高くなる．スキルを多く持たないエージェントが参加と辞退を多く行ったためにこのような結果が得られた．タスク量が $[5.0, 8.0]$ の範囲では，ソフトウェアを辞退したエージェントは少ない．式 (5) より，ソフトウェアのタスク量が多い場合，多くのエージェントが参加しやすくなる．そのため，エージェント 1 個あたりのタスク量が小さくなり，辞退するエージェントが少なくなる．タスクがある量を越えた場合，エージェントが結託して開発に取り組むと考えられる．

7　利他行動を適用したシミュレーション結果

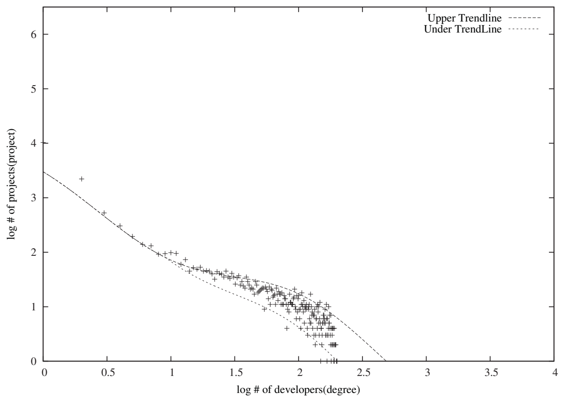
エージェントの行動に利他行動を適応したシミュレーション結果を Fig. 3(a) に示す．横軸はソフトウェアのエージェント数の対数，縦軸はソフトウェア数の対数である．

Fig. 2(a) と比較して，エージェント数の対数が 0.0 ∼ 0.5 においてはソフトウェアの数が減少し，エージェントの数の対数が 0.5 ∼ 1.5 においてはソフトウェアの数が上昇している．この結果より，利他行動によってエージェントが参加することでコミュニティの成長が見られたと言える．

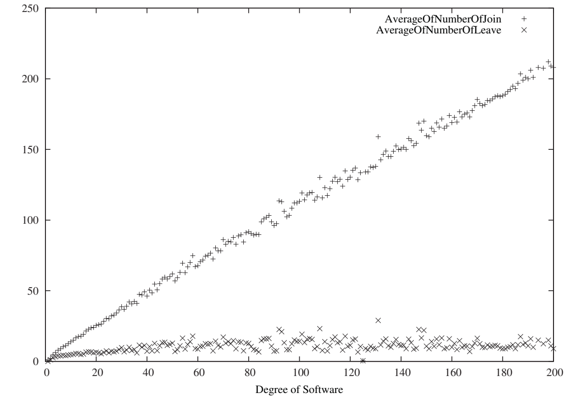
7.1　シミュレーション結果の比較

利他行動なしのシミュレーション結果と利他行動ありでのシミュレーション結果の比較する．比較する項目として，全体のソフトウェアの開発に要する時間の平均と，タスク量と辞退したエージェントの数の関連性である．

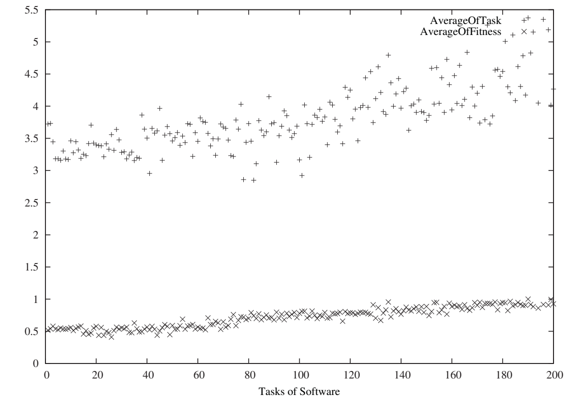
開発効率について，利他行動を適用していない場合と適用した場合のシミュレーション結果を以下の Table 1 に示す．ここでは，ソフトウェアの開発効率に関して，シミュレーションを通して生成されたソフトウェ



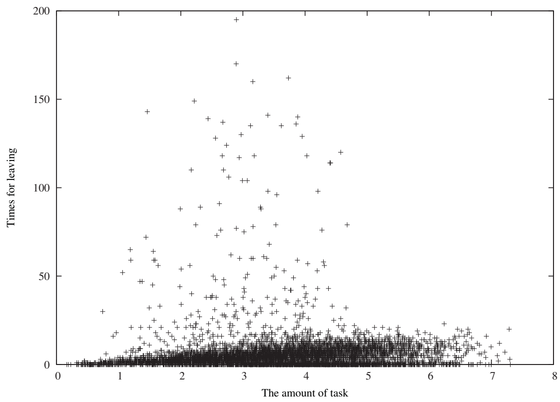
(a) ソフトウェアに参加しているエージェント数とソフトウェアの数の関連性



(b) ソフトウェアに参加しているエージェント数とエージェントの行動の関連性



(c) ソフトウェアに参加しているエージェント数とソフトウェアのタスク量と適応度の関連性



(d) タスク量と辞退したエージェント数の関連性

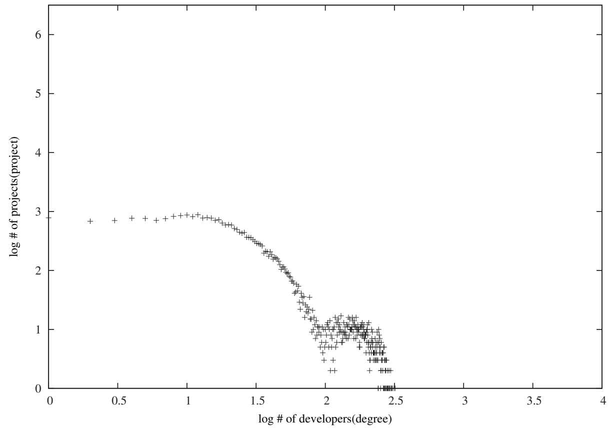
Fig. 2: シミュレーション結果

アのうち，開発に参加しているエージェントが 1 個でも存在するソフトウェア数とその開発時間の平均を示している．開発時間の平均を比較すると，‘**利他行動あり**’ の場合では，‘**利他行動なし**’ の場合の半分以下であることがわかる．利他行動によってより開発に適したコミュニティが形成されたために，このような結果が得られたと考えられる．

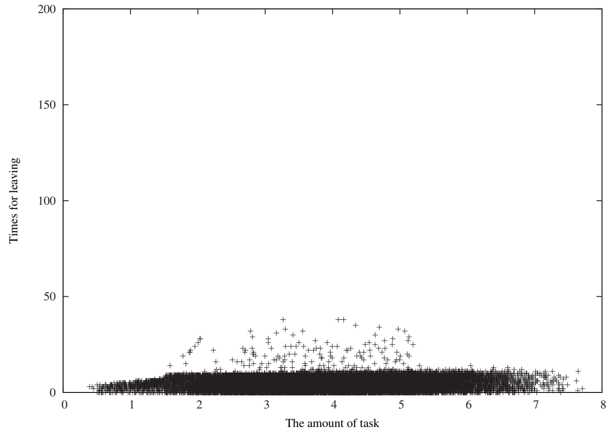
次に，ソフトウェアのタスク量と辞退したエージェントの数の関連性について検討する．Fig. 2(d) と Fig. 3(b) は横軸はソフトウェアのタスク量，縦軸は辞退したエージェント数であり，Fig. 2(d) は ‘**利他行動なし**’ のシミュレーション結果，Fig. 3(b) は ‘**利他行動あり**’ のシミュレーション結果である．‘**利他行動なし**’ の結果 (Fig. 2(d)) では，タスク量が 1.5 ~ 5.0 のソフトウェアを辞退したエージェントが多いことを表している．これらのソフトウェアに関して，‘**利他行動あり**’ の結果 (Fig. 3(b)) では，辞退したエージェント数が減少している．利他行動によってタスクが分散されるため，辞退したエージェント数が減少したと考えられる．利他行動を適用した場合，開発に参加しているエージェントが 1 個でも存在するソフトウェアの数は増加し，開発時間の平均は減少している．これは，利他行動によって辞退したエージェントの数が減少したため，エージェントが 1 個も参加していていないソフトウェアが減少したと考えられる．

Table 1: ソフトウェアの開発時間の平均

	ソフトウェアの数	開発時間の平均
利他行動なし	6555	7.70
利他行動あり	26119	3.74



(a) ソフトウェア開発に参加しているエージェント数とソフトウェア数の関連性



(b) ソフトウェアのタスク量と辞退したエージェント数の関連性

Fig. 3: 利他行動を取り入れたシミュレーション結果

謝辞

本研究の一部は，JSPS 科研費 25330366 の助成を受けたものである．

8 おわりに

ソフトウェア開発において，実際に SourceForge.net のプロジェクトなどのデータを取得し，分析結果をもとに開発のタスクやエージェントのスキルを取り入れた環境を開発し，エージェントの行動に利他行動を取り入れたシミュレーションを行った．利他行動を取り入れていない場合の結果と比較して，ソフトウェア開発に要する時間が半分程度に減少する結果となった．ソフトウェア開発において利他行動を取り入れることで，全体としての開発効率が向上したと言える．

今後は，ソーシャル活動に対する応用事例として，Facebook の投稿におけるソーシャルコミュニティに関して，利他行動によるコミュニティの成長を促進する手法の提案を行う．

参考文献

1) Madey, G., Gao, Y., Freeh, V., Tynan, R., Hoffman, C.: ‘Agent-based Modeling and Simulation of Collaborative Networks’, Proc. of Americas Conference on Information System(AMCIS2002) pp.1839-1842(2003)

2) `https://sourceforge.net/` (2015/06/22)

3) 上本拓也，市村匠，”オープンソースソフトウェア開発におけるネットワーク成長モデルに基づくソーシャルコミュニティの形成法の提案”，2015 IEEE SMC Hiroshima Chapter Young Researchers WorkShop, pp.143-146(2015)

4) T, Ichimura, and T, Uemoto, ”Analysis of the Social Community Based on the Network Growing Model in Open Source Software Community”, Proc. of IEEE 8th International Workshop on Computational Intelligence and Applications (IWCIA2015), pp.149-153(2015)

5) Takumi Ichimura, Takuya Uemoto, Akira Hara, and Kenneth J. Mackin, ”Emergence of Altruism Behavior in Army Ant Based Social Evolutionary System”, SpringerPlus, A Springer Open Journal, Vol.3, 712, doi:10.1186/2193-1801-3-712,(2014).

6) `http://srda.cse.nd.edu/mediawiki/index.php/Making_Queries` (2015/06/22)

7) T, Ichimura. T, Uemoto. A, Hara.: ’Emergence of Altruism Behavior for Multi Feeding Areas in Army Ant Social Evolutionary System’, Proc of IEEE SMC 2014, San Diego, CA, pp170-175(2014)

8) T, Gao. G, Madey. V, Freeh.: ”Modeling and Simulation of the Open Source Software Community”, Agent-Directed Simulation Conference, San Diego, CA, April(2005)

9) Barabasi, A.L. Albert, R.: ‘Emergence of scaling in random networks’, Science 286, October 15 1999, pp509-512(1999)

相互通信と外界センサを持たないロボット少数台が示す “知的” 挙動

○伊丹哲郎 （ロボット産業振興会議）

‘Intelligent’ Behavior By A Few Robots Without Apparatuses for Mutual Communication

∗ITAMI Teturo (Robotics Industry Development Council)

Abstract– A group of a few macroscopic robots with neither apparatuses for mutual communication nor external sensors is shown to give an intelligent behavior. Four($N_0 = 4$) robots make an ob ect track an required path in a plane region surrounded by walls. This is done by changing our viewpoint of describing purely Newtonian systems from mechanical one to number density picture even for systems with a few robots.

Key Words: Intelligent behavior, Robots with neither apparatuses for mutual communication nor external sensors, Group robots with a few robots, Boltzman distribution, Broadcast control

1 はじめに

本研究は，スォーム・ロボティクス [5] の視点から知的挙動の創発を図るもの，と位置づけられる．ロボットは「巨視的」であって ~ 1[m]，~ 1[kg] のオーダの大きさを持ち，~ 1[s] オーダの時間で仕事をするシステムを対象とする．ロボットが群れをなすシステムについて，これまでさまざまな研究があるが，そこではロボット間の相互作用が強調されることが多い．さらに一般にロボティクスにおいては,「衝突しない」ことが目標の一つをなす．これら相互作用や衝突フリーは，相互通信や外界を知るセンサのロボットへの搭載を前提とする． それでは，できる だけ 少数個の簡素な ロボットを集めたシステムを考えることに，意味があるだろうか．ロボットの台数は，多くても ~ 100 台ていどに抑えたい．ロボットは「簡素」に作ってあるため，外界センサによる環境の認識などはしない．ましてや互いの通信は，想定さえしない． こんなロボットを集めたとして，一体何ができるのか，という疑問については，ブラウン運動やブラウン・モータ [1] にヒントがある． われわれの簡素ロボット群とブラウン運動 (あるいはブラウン・モータ) の比較は次の通りである．

- ロボット ⇔ 液体分子
- 物体 ⇔ 被駆動部 (花粉粒子相当)

すなわちロボットを液体分子のように動かすことができれば，物体を花粉のように意のままに操れるはずである．そこで本研究でも，モノをある決まった軌道に沿って動かすこと，を簡素ロボット群の仕事の目標とする．しかしわれわれのロボットが巨視的，ブラウン運動は微視的という自明な差のほかに，両システムは次の点で大きく異なる．

- いくら多くても 10^4 台？⇔$\sim 10^{23}$ 個の液体分子
- 温度場で，ロボットは動ない ⇔ 液体分子は動く
- 純粋な古典力学系 ⇔ 確率システム

この第２，３項に列挙した差について，われわれは以下の左に示した項目を設定してきた [2, 3]．

- ポテンシャル信号 ⇔ 温度場
- リウビル方程式 ⇔ ランジュバン方程式

すなわち図 **1** に示した通り，ポテンシャル信号 V_{cnt}(の

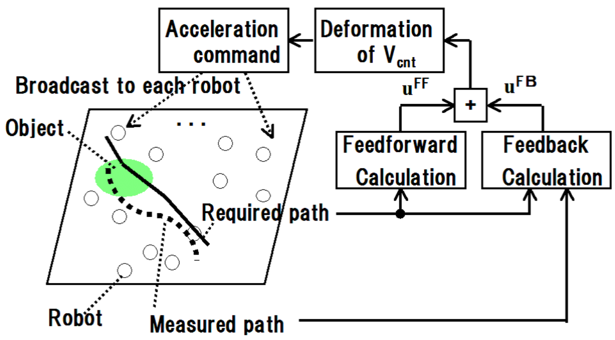


Fig. 1: Conceptual design.

位置による偏微分) を各ロボットにブロードキャストする構成をとった．ここで V_{cnt} は物体の希望軌道 $\vec{X}^{req}(t)$ に応じたフィードフォワード u^{FF} と，計測される軌道 $\vec{X}$ の偏差によるフィードバック u^{FB} の和とした．またロボット群はニュートン方程式に厳密に従う古典力学系であるため，これをリウビル方程式の形に変換した．すわわち，ロボット群を各ロボットの位置，速度ではなく，ある範囲の位置，速度を取るロボットの個数密度で記述した．個数密度としては，統計力学的な平衡分布を使うものとした．

しかしロボットの個数については，これまで明確な議論をしていない．そこで本稿では，衝突エネルギーの交換過程の分析により，ロボット群の平衡分布

$$f_v = C \cdot \exp\left(-\tilde{\beta} E\right) \tag{1}$$

による記述の妥当性を，ロボット少数台に対して確認する．なお (1) で，$\tilde{\beta}^{-1}$ は指数分布の形を特徴づけるパラメータでエネルギーの次元を持ち「温度」と呼ぶ．また E はシステムのエネルギーであり，C はシステムのロボット台数を決める比例定数である．今後 (1) をボルツマン分布と呼ぶ．

結果としてフィード・フォワードを (1) で計算でき，4 台でも物体運動を制御できる．これにより，分布 (1) の制御により，簡素なロボットを活用した搬送システムが可能となる．

はじめに **2** で，少数系のボルツマン分布への接近のタイミングを確認する．次いで，ブラウン運動にヒントを得た，巨視的な物体を動かす群ロボットのシステムを，これまでの研究 [2, 3] のまとめとして **3** で整理する．シミュレーションは **4** で与え，特定の $\vec{X}^{req}(t)$ のトラッキングについてのフィード・フォワードの制御成績と，その PD フィードバック補償を示す．まとめと今後の課題は **5** で述べる．

2　少数系の統計力学

統計力学は，ありとあらゆる可能な状態が等しい確率で出現する，とする「等重率の原理」に基礎づけられている．そしてボルツマン分布 (1) は統計力学の結果である．であるとすれば，あらゆる可能な状態が出尽くして等重率が成り立つ段階になってはじめて，(1) 式を使える，と考えるのが論理的である．しかし文献 [4] によれば，多数の個体のあいだでのエネルギー交換システムでは，ボルツマン分布は等重率が成立する遥か前に成立する．すなわち，時間的な推移の順序は

1. 集団にわたるボルツマン分布
2. 各個体のボルツマン分布
3. 等重率

となる．さらに個体が N 個であってこれらが M 個のエネルギー塊を交換するときは，等重率の原理からエネルギー塊を m 個もつような個体の数の分布が

$$f(m) = \frac{{}_{N-1}C_{M-m}}{{}_{N}C_{M}} \tag{2}$$

と計算される．そしてこの (2) の分布は M，N が ~ 5 ていどであっても十分に指数関数の分布 $C \times e^{-\tilde{\beta} m}$ でフィッティングできるのである．指数関数フィッティングと等重率から導かれる分布との誤差は，たとえば $N = 4$ エージェントで $M = 3$ チップを交換するときでも 1.2[%] に過ぎない．従って少数系のエネルギー分布に対しても，(1) 式を用いてよい．

以下で具体的な計算をして，文献の主張を確認する．続いてわれわれのロボット群にこの主張を適用し，(1) を使う妥当性を確認する．

はじめに，図 **2** に示したエネルギー交換の原型モデル を研究する．われわれは初期にロボットを平面上で

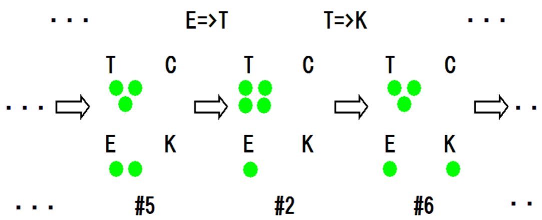


Fig. 2: Exchanging $M = 5$ tips among $N = 4$ agents.

物体から見て東西南北のどこかに配置するとして，その最小限の $N_0 = 4$ を本稿では研究する．このため原型モデルでも，エネルギーを授受するエージェントは $N = 4$ とする．またエネルギー塊はいくつでも良いが，可能なエネルギー範囲を分割する切りのよい値として $M = 5$ 分割，つまり $M = 5$ 個のチップが交換される，と設定する．図 2 のシステムに出現するあらゆる状態を書き下すと，表 **1** の通り 56 個ある．それは，エージェント T が $M = 5$ チップを独占する状態 #1 から，C が独占する #56 までである．ちなみに図 2 では，なんらかの初期分配から出発して状態 #5 まで来ている．ここで，たとえばサイコロを振るなりして E から T にチップを渡すことになったとすると，結果は状態 #2 である．さらに次いで T から K にチップが渡ったとすると状態は #6 に推移している．このように状態は推移していく．しかしすべての 56 状態が平等に出現するには，相当の回数の授受プロセスが必要であろうことは想像に難くない．状態の出現頻度の標準偏差を平均値に対して図 **3** の青実線で示した．すると，等重率が成り立つ段階では標準偏差がゼロ，つまりその対数は負の無限大とならねばならない．しかしたかだか 200 回のチップ授受では，その $-\infty$ になるべき偏差 (と平均値の比の log) はせいぜい ~ -0.5 でしかない．すなわち，状態の出現頻度はまったく不平等なままである．ただ青線はゆるやかに下降しており，非常に遠い将来には晴れて等重率が成立と予想される．しかしこの早い段階ですでに，エネルギー分布はほぼフィッティング値 $1.632 \times e^{-0.488 \times m}$ に達していて，緑線は横軸に平行に推移している．これは文献の主張を裏付けるものである．

state	T	E	K	C
1	5	0	0	0
2	4	1	0	0
3	4	0	1	0
4	4	0	0	1
5	3	2	0	0
6	3	1	1	0
...	...	...	...	...
54	0	0	2	3
55	0	0	1	4
56	0	0	0	5

Table 1: $M = 5$ resources among $N = 4$ agents

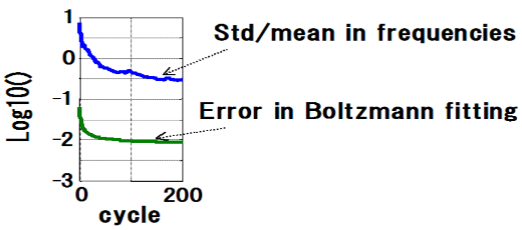


Fig. 3: Error of *apriori equal probability* and that of Boltzmann distribution fitting for $M = 5$ tips exchange among $N = 4$ agents.

続いて実際の $N_0 = 4$ 個のロボット群で，エネルギー範囲を $M = 5$ に分割した系の結果は図 **4** の通りである．ここでは分子動力学系として計算しているので，壁も物体もない．ゆえに互いの衝突でのみ，ロボット間でエネルギー交換がなされる．すなわち緑破線が図 2 の

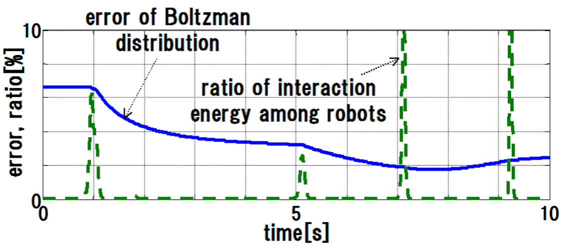


Fig. 4: Distributing kinetic energy in 5 bins among 4 robots.

チップ交換のタイミングに対応する．ここでは数回のエネルギー交換で，すでにボルツマン分布への接近がはかれている．実システムでは壁も物体もあって，これを介したロボット間のエネルギー交換がより頻繁に起こる．すなわち (1) 式への接近性がより早いと予想される．

3　巨視的なブラウン運動

ブラウン運動に着想を得たわれわれの群ロボットのシステムにあっては，液体分子としてのロボットが花粉粒子としての物体を動かす．ロボットも物体も巨視的であるから，そこでは物理過程のすべてを古典力学で完全に記述できる．しかし物体位置 $\vec{X}(t)$ の設定値 $\vec{X}^{req}(t)$ へのトラッキングをポテンシャル力 $V_{cnt}(\vec{x};t)$ の条件として書き下すことは，原理的に困難である．それは，ニュートン力学はロボット群，物体すべての初期位置と速度の情報を必要とするからである．またたとえそれら初期条件の詳細が分かったとしても，ポテンシャル力を書き下すことは，ロボット数が増えれば，困難である．そこでわれわれは

- ロボット群の状態をボルツマン分布で近似し，
- ロボット群から物体への衝突力 $\vec{F}_{col}$ を具体的に計算する．
- その $\vec{F}_{col}$ 作用の下で，物体はニュートン力学に従って動く，

と考える．このようにすれば，物体のトラッキング条件は容易にポテンシャル力の条件として書き直せる．

3.1　ニュートン方程式

位置 $\vec{x}_R$ にあるロボット R には，速度 $\vec{v}_R$ に比例するマサツ力 $-\gamma\vec{v}_R$ とともに，制御ポテンシャル $V_{cnt}(\vec{x}_R;t)$，物体との衝突 $V_B(|\vec{x}_R-\vec{X}|)$，壁反力 $V_{R0}(\vec{x}_R)$ および $\vec{x}_{R'}$ にある別のロボット R' との衝突のポテンシャル $V_{col}(|\vec{x}_R-\vec{x}_{R'}|)$ が作用する．そこでロボットの外場ポテンシャルを

$$V(\vec{x};\vec{X};t) \equiv V_{cnt}(\vec{x};t)+V_{R0}(\vec{x})+V_B(|\vec{x}-\vec{X}|) \quad (3)$$

とまとめると，ロボットは運動方程式

$$\begin{aligned} m\dot{\vec{v}}_R &= -\frac{\partial}{\partial\vec{x}_R}V(\vec{x}_R;\vec{X}(t);t) \\ &\quad -\frac{\partial}{\partial\vec{x}_R}\sum_{R'\neq R}V_{col}(|\vec{x}_{R'}-\vec{x}_R|)-\gamma\vec{v}_R \end{aligned} \quad (4)$$

に従う．一方，位置 $\vec{X}$ の物体には V_B によるロボットからの衝突力

$$\vec{F}_{col} = -\frac{\partial}{\partial\vec{X}}\sum_{R=1}^{N_0}V_B(|\vec{x}_R-\vec{X}|) \quad (5)$$

とポテンシャル V_{B0} で記述される壁からの反力およびマサツ力 $-\delta\vec{V}$ が作用する．そこで物体の運動方程式は，速度 $\vec{V}$ として

$$M\dot{\vec{V}} = \vec{F}_{col}-\frac{\partial}{\partial\vec{X}}V_{B0}(\vec{X})-\delta\vec{V} \quad (6)$$

である．

運動方程式 (4) と (6) は，速度の定義 $\dot{\vec{x}}=\vec{v}$ と $\dot{\vec{X}}=\vec{V}$ を含めて制御理論の意味での状態方程式をなす．1 個の物体とロボット N_0 台の 2 次元位置，速度が $2\times2\times(1+N_0)$ 次元の状態変数 $\vec{\mathcal{X}}$ である．また観測方程式は，物体の位置のみが計測対象であるため，$2\times2\times(1+N_0)$ 次元の状態ベクトル $\vec{\mathcal{X}}$ を 2 次元の $\vec{X}$ に行列 C で射影する $\vec{X}=\mathsf{C}\vec{\mathcal{X}}$ になる．われわれはロボット群を，ポテンシャル $V_{cnt}(\vec{x};t)$ を時々刻々に変形することで運動させ，ロボット群が物体に衝突することで出力 $\vec{X}$ を希望する $\vec{X}^{req}$ にトラックする．そこで操作入力 u は V_{cnt} の「形」であって，これが作用するのはロボットの速度 $\vec{v}_R$ であり物体の速度ではないこと，を改めて注意しておく．なお制御ポテンシャル V_{cnt} のあらわな時間 t への依存性は，ゆいいつこの入力 u を介したものである．

以上の制御スキームにあって問題となるのは，以下である．

- 時間関数 $\vec{X}^{req}(t)$ を与えた下で，適切な時間関数 u が存在するかは自明でない．
- さらに理論の上で存在したとしても，状態方程式が 1 階常微分方程式であることにより，この u は各ロボットの位置と速度の初期値に依存する．しかし計測するのは物体の位置 $\vec{X}$ のみであり，ロボットの初期配置は不明である．
- またロボット台数 N_0 が増えると u の計算が複雑さを増すことは言うまでもない．

そこで本稿では，u を制御論的な状態方程式に基づいて計算することは避ける．具体的に線形の V_{cnt} を対象として，統計力学的手法により具体的に u を計算する．しかし任意の時間関数 $\vec{X}^{req}(t)$ を設定できるか，は自明でない．また物体個数を本稿では 1 としたが，これが複数台ならどうなるか？これらは制御理論的に分析する必要があり，将来の課題とする．

3.2　リウビル方程式と平衡分布

ロボットの運動方程式 (4) は，ラグランジアン

$$\begin{aligned} L &= e^{\frac{\gamma}{m}t}\{\sum_{R=1}^{N_0}(\frac{m}{2}\vec{v}_R^2-V(\vec{v}_R;\vec{X};t) \\ &\qquad -\sum_{R'\neq R''}V_{col}(r_{R'R''}))\} \end{aligned} \quad (7)$$

から導くこともできる．正準運動量を公式通りに計算すれば，

$$\vec{p}_R = \frac{\partial L}{\partial \vec{v}_R} = e^{\frac{\gamma}{m}t} m \vec{v}_R \tag{8}$$

次のハミルトニアンによる正準方程式がロボット系を記述する．

$$\begin{aligned} H &= \sum_{R=1}^{N_0} \vec{v}_R \cdot \vec{p}_R - L \\ &= \sum_{R=1}^{N_0} H_0 + H_{int}(\vec{x}_1, \cdots, \vec{x}_{N_0}) \end{aligned} \tag{9}$$

このハミルトニアンが自由パート

$$H_0 = e^{-\frac{\gamma}{m}t} \frac{\vec{p}^2}{2m} + e^{\frac{\gamma}{m}t} V(\vec{x}; \vec{X}; t) \tag{10}$$

と衝突パート H_{int} の和に分離できることから，個数密度を表す次の関数

$$f(\vec{x}_1, \vec{p}_1, \cdots, \vec{x}_{N_0}, \vec{p}_{N_0}; t) \tag{11}$$

を用いてリウビル方程式を具体的に設定できる．すなわち 2 体分布関数を

$$f_2(\vec{x}_{N_0-1}, \vec{p}_{N_0-1}, \vec{x}_{N_0}, \vec{p}_{N_0}; t) \equiv \int d\vec{x}_1 \cdots \int d\vec{p}_{N_0-2} f \tag{12}$$

また 1 体分布関数を

$$f_1(\vec{x}_{N_0}, \vec{p}_{N_0}; t) \equiv \int d\vec{x}_{N_0-1} d\vec{p}_{N_0-1} f_2 \tag{13}$$

として，リウビル方程式が次になる．

$$\begin{aligned} \frac{\partial f_1}{\partial t} + [f_1, H_0] &= -(N_0 - 1) \\ &\quad \times \int d\vec{x}_{N_0-1} d\vec{p}_{N_0-1} [f_2, V_{col}(r_{N_0, N_0-1})] \end{aligned} \tag{14}$$

(14) 右辺をゼロとできるのであれば，十分に遅い時間変化を想定したうえで，

$$f_1^{eq}(\vec{x}, \vec{p}; t) \equiv C_1 e^{-\beta H_0(\vec{x}, \vec{p}; t)} \tag{15}$$

という平衡分布が (14) を満たす．計測できるのは運動量 $\vec{p}$ でなく速度であるから，

$$\vec{v} = e^{-\frac{\gamma}{m}t} \frac{\vec{p}}{m} \tag{16}$$

によりハミルトニアンを書き直しておく．自由パートは

$$\begin{aligned} \tilde{H}_0(\vec{x}, \vec{v}; t) &\equiv e^{-\frac{\gamma}{m}t} H_0(\vec{x}, \vec{p}; t) \\ &= \frac{m}{2} \vec{v}^2 + V(\vec{x}; \vec{X}; t) \end{aligned} \tag{17}$$

であり，比例定数と温度パラメータを書き直して，

$$C_v(t) \equiv m^2 e^{2\frac{\gamma}{m}t} C_1 \tag{18}$$

$$\tilde{\beta}(t) \equiv e^{\frac{\gamma}{m}t} \beta \tag{19}$$

平衡分布は

$$f_v^{eq}(\vec{x}, \vec{v}; t) = C_v(t) e^{-\tilde{\beta}(t) \tilde{H}_0(\vec{x}, \vec{v}; t)} \tag{20}$$

になる．

ふたつの未知パラメータ C_v および $\tilde{\beta}$ は，ロボット台数が N_0 であり全エネルギーが E_{tot} であること，の２条件から計算される．ロボット台数 N_0 の計算は

$$N_0 = \int d^2\vec{v} \int d^2\vec{x} f_v(\vec{x}, \vec{v}; t) \tag{21}$$

であり，全エネルギーは

$$E_{tot} = \int d^2\vec{v} \int d^2\vec{x} f_v(\vec{x}, \vec{v}; t) \tilde{H}_0(\vec{x}, \vec{v}; t) + E_{Obj} \tag{22}$$

である．ここで (22) 中の E_{Obj} は，物体の運動エネルギーと物体への壁反発エネルギーの和である．計算の便宜のため $I_v[\tilde{\beta}(t)] \equiv \int d^2\vec{v} e^{-\tilde{\beta}(t)\frac{m}{2}\vec{v}^2}$ と $I_x[\tilde{\beta}(t); t] \equiv \int d^2\vec{x} e^{-\tilde{\beta}(t) V(\vec{x}; \vec{X}(t); t)}$ を導入する．ロボット台数は

$$N_0 = C_v I_v I_x \tag{23}$$

また全エネルギは，$\tilde{\beta}$ による偏微分を $' \equiv \frac{\partial}{\partial \tilde{\beta}}$ と略記して，

$$E_{tot} = C_v(-I_v^{'}) I_x + C_v I_v (-I_x^{'}) + E_{Obj} \tag{24}$$

と計算される．さらに (24) を (23) で除すると，

$$f(\mathsf{u}, \tilde{\beta}) = \frac{-I_v^{'}}{I_v} + \frac{-I_x^{'}}{I_x} \tag{25}$$

を定義しておけば

$$E_{tot} = N_0 \times f(\mathsf{u}, \tilde{\beta}) + E_{Obj} \tag{26}$$

である．すなわち (25) の f は，ロボット 1 台あたりに分配されたエネルギーである．

時間初期 t_0 では物体の運動方程式 (6) にポテンシャル形状 $\mathsf{u}(t_0)$ と物体の位置 $\vec{X}(t_0)$，速度 $\vec{V}(t_0)$ を代入した式と，$E_{tot}(t_0)$ と $E_{Obj}(t_0)$ が分かっている (26) とを連立して $\mathsf{u}(t_0)$ と $\tilde{\beta}(t_0)$ が計算できる．その $\tilde{\beta}(t_0)$ の結果を (23) に代入して $C_v(t_0)$ を得る．これらの連立式は 3 式であるから，ポテンシャル形状 $\mathsf{u}(t_0)$ が n 個の未知パラメータで決まるとすると，それらに対する 2 個の関係式を強制することになる．

しかし物体の運動を制御するのはポテンシャル $V_{cnt}(\vec{x}; t)$ の時々刻々の操作であり，全エネルギは時間変化する．さらにマサツによるエネルギ散逸もあり得る． すなわちエネルギが与えられる初期以降の任意時点 t では全エネルギ E_{tot} の値は不明である．そこで初期値を与える 1 階の常時間微分方程式で温度パラメータ $\tilde{\beta}$ を決めることにする．このために先ず運動方程式 (4) と (6) を使って，全エネルギの時間微分を計算する．

$$\frac{dE_{tot}(t)}{dt} = -\gamma \sum_{R=1}^{N_0} \vec{v}_R^2 - \delta \vec{V}^2 + \frac{\partial}{\partial t} \sum_{R=1}^{N_0} V_{cnt}(\vec{x}_R; t) \tag{27}$$

物体とロボットの衝突エネルギ $V_B(|\vec{x}-\vec{X}|)$ は (27) の計算プロセスで相殺する．全エネルギの統計平均 (26) の時間微分が，時間微分 (27) の統計平均に等しい，$\frac{d}{dt}\overline{E_{tot}} = \overline{\frac{dE_{tot}}{dt}}$，と仮定すれば，

$$N_0\frac{df}{dt} + \frac{dE_{Obj}}{dt} = -\frac{2\gamma}{m}N_0\frac{1}{\tilde{\beta}(t)} - \delta\vec{V}^2 + N_0\overline{\frac{\partial V_{cnt}}{\partial t}} \quad (28)$$

を得る．これは

$$\frac{df}{dt} = \frac{\partial f}{\partial \mathsf{u}}\dot{\mathsf{u}} + \frac{\partial f}{\partial \tilde{\beta}(t)^{-1}}\dot{\tilde{\beta}}(t)^{-1} \quad (29)$$

を使うことで，$\tilde{\beta}(t)^{-1}$ の常微分方程式に導く．これを初期値 $\tilde{\beta}(0)^{-1}$ の下で時間積分して，任意時点の温度パラメータが計算される．もちろん $\tilde{\beta}^{-1}(t)$ の計算のためには，操作入力 $\mathsf{u}(t)$ の計算が平行して行われていることが前提である．

物体の運動方程式は

$$M\dot{\vec{V}} = \vec{F}_{col}(\vec{X},\vec{V};t) - \frac{\partial}{\partial \vec{X}}V_{B0}(\vec{X}) - \delta\vec{V} \quad (30)$$

であり，質量 M と加速度 $\dot{\vec{V}}$ の積はロボットによる衝突力 $\vec{F}_{col}$ と壁反力 $-\frac{\partial}{\partial \vec{X}}V_{B0}$ およびマサツ力 $-\delta\vec{V}$ の和に等しい．ポテンシャル力は形のパラメータ u で決まる．このパラメータを温度パラメータ $\tilde{\beta}^{-1}$ と同時に各時点 t で決める．

実際には

$$V_{cnt}(\vec{x};t) = \alpha_1(t)x_1 + \alpha_2(t)x_2 \quad (31)$$

と線形に選ぶ．このとき $\frac{\partial V_{cnt}}{\partial \vec{x}} = \vec{\alpha}$ となり，これはロボット位置 $\vec{x}$ に依存しない．すなわちロボットに位置センサを搭載する必要がない．この $\vec{\alpha}$ は，物体の 2 成分の運動方程式 (30) を 1 元の，(29) と連立し，$\tilde{\beta}^{-1}$ と同時に計算する．

3.3 衝突による力 $\vec{F}_{col}$ の計算

分布関数を使って，ここでは物体に作用するロボットの衝突力を計算する．強調されるのは，物体がロボットに対する外場ポテンシャル $V(\vec{x};\vec{X}(t);t)$ の正勾配に沿って動く，ことである．ロボットが V の負勾配に沿って動いていることは断るまでもない．

ロボットが平衡分布にあると仮定して，ロボットが及ぼす物体への衝突力を計算する．物体への力は図 **5** の配位に従って計算できる．時間 dt の間に進む衝突過程を考える．ロボットが物体に及ぼす力積が $Md\vec{V}'$，位置 $\vec{x}$，速度 $\vec{v}$ のロボットが速度幅 $d^2\vec{v}_r$ に $d^2N = N_0 \times f_v^{eq}(\vec{x},\vec{v};t)d^2\vec{v}_r$ 個，その存在領域は $dS = D \times L$ の平行四辺形の中，とすると，ロボット群が物体に与える力は次で計算できる．

$$\vec{F}_{col} = \frac{1}{dt}\int_{\theta=0}^{2\pi}\int_{v_r=0}^{\infty}\int_{\phi=\frac{\pi}{2}}^{\frac{3\pi}{2}} Md\vec{V}'d^2NdS \quad (32)$$

以下で各項の詳細説明を行い，特に $\vec{V}=0$ でどのような力の式になるか，に注目する．まず衝突の相対速度ベクトル $\vec{v}_r$ を絶対値 v_r と角度 ϕ でパラメトライズする．ベクトル $\vec{\theta}_{x1}$ を，物体中心から衝突点 $\vec{x}$ への方向ベクトルとしよう．角度 ϕ を，ベクトル $\vec{\theta}_{x1}$ に平行にとった $\vec{\theta}/\!/_{x_1}$ から測る．範囲 $\frac{\pi}{2} < \phi < \frac{3\pi}{2}$ だけの ϕ で衝突が起こり，これは負値の $\cos\phi$ に導くことに注意しよう．衝突の反発係数を e とすれば，ロボット衝突に起因する物体速度の増分は

$$\begin{aligned} d\vec{V}' &= -|d\vec{V}'|\vec{\theta}_{x1} \\ &= -\frac{(1+e)(-v_r\cos\phi)}{1+\frac{M}{m}}\vec{\theta}_{x1} \end{aligned} \quad (33)$$

である．この (33) で増分 $d\vec{V}'$ が $\vec{\theta}_{x_1}$ と逆方向であることを強調した．単位面積の中には

$$\begin{aligned} d^2N &= f_v(\vec{x},\vec{v};t)d^2\vec{v}_r \\ &= C_v e^{-\tilde{\beta}\left(\frac{1}{2}m\vec{v}^2+V(\vec{x};\vec{X};t)\right)}d^2\vec{v}_r \end{aligned} \quad (34)$$

個のロボットが有る．図 5 を見れば $|\vec{x}-\vec{X}| = R_B$ と

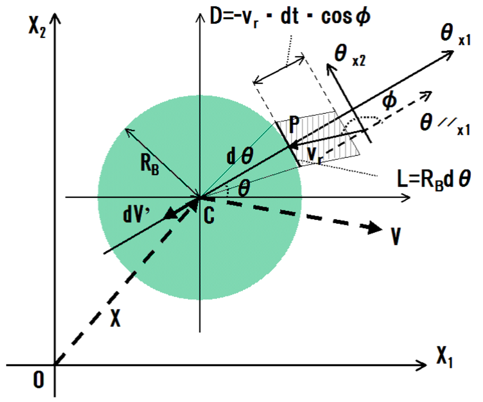


Fig. 5: Collision process of object with robots in statistical mechanics.

分かるが，このため (34) の指数関数で $e^{-\tilde{\beta}V_B(|\vec{x}-\vec{X}|)}$ は正の定数 $c_B = e^{-\tilde{\beta}V_B(R_B)}$ になっている．関数

$$g(\vec{x};t) \equiv e^{-\tilde{\beta}(V_{cnt}(\vec{x};t)+V_{R0}(\vec{x}))} \quad (35)$$

を使って (34) を簡潔にあらわすと，

$$d^2N = C_v \times c_B \times g(\vec{x};t)e^{-\tilde{\beta}\frac{1}{2}m\vec{v}^2}d^2\vec{v}_r \quad (36)$$

である．注意すべきは d^2N があきらかに正であることである．これらのロボットの相対速度は $\vec{v}_r \sim \vec{v}_r + d\vec{v}_r$ である．時間間隔 dt のあいだに物体の線要素 $L = R_Bd\theta$ には

$$dS = L \times D = R_Bd\theta(-v_r\cos\phi)dt \quad (37)$$

にあるロボットだけが物体に衝突できる．この面積要素 dS は，$\cos\phi$ が負であるため，正値をとる．個数 $d^2N \times dS$ だけのロボットそれぞれが，物体に力積 $Md\vec{V}'$ を与える．この力積を，全方向 $\theta = 0 \sim 2\pi$ にわたり，相対速度の絶対値の可能な値すべてについて，なおかつ $\phi = \frac{\pi}{2} \sim \frac{3\pi}{2}$ の範囲で積分すると，ロボットから物体に及ぼす力が得られる．ロボットの $\vec{v}_r$ が物体の速度 $\vec{V}$ に対する相対速度であることから，この衝突力は $\vec{V}$ に

も依存する．

$$\vec{F}_{col}(\vec{X},\vec{V};t) = -\frac{1}{dt}\int_{\theta=0}^{2\pi}\int_{v_r=0}^{\infty}\int_{\phi=\frac{\pi}{2}}^{\frac{3\pi}{2}} M\frac{(1+e)(-v_r\cos\phi)}{1+\frac{M}{m}}\vec{\theta}_{x1} \times C_v c_B g(\vec{x};t)e^{-\tilde{\beta}\frac{1}{2}m\vec{v}^2}d^2\vec{v}_r \times R_B d\theta(-v_r\cos\phi)dt \quad (38)$$

この積分を，特に物体が静止 ($\vec{V}=0$) している場合について，実行すると興味深い結果になる．量 $e^{-\tilde{\beta}\frac{m}{2}v_r^2}$ と $\cos^2\phi$ を含む積分に公式 $d^2\vec{v}_r = v_r dv_r d\phi$ を適用すると，関数 $g(\vec{x};t)\vec{\theta}_{x1}$ に負の定数 $-c$ をかけたものが残る．この負であることは $d\vec{V}' = -|d\vec{V}'|\vec{\theta}_{x1}$ のマイナス号に起因している．引き続いて $-c\times g(\vec{x};t)\vec{\theta}_{x1}$ を θ にわたり積分するときには，衝突点では $\vec{x}=\vec{X}+R_B\vec{\theta}_{x1}$ であることに注意する．計算を進めるにさいして，関数 g を級数展開しておく．$g(\vec{x};t)=g(\vec{X};t)+R_B\vec{\theta}_{x1}\cdot\frac{\partial g}{\partial\vec{X}}+O({R_B}^2)$．任意のベクトル $\vec{a}$ に対して $\int_0^{2\pi}d\theta\vec{\theta}_{x1}=0$ と $\int_0^{2\pi}d\theta(\vec{\theta}_{x1}\cdot\vec{a})\vec{\theta}_{x1}=\pi\vec{a}$ が成り立つから，最終的に積分 (38) は，

$$\begin{aligned}\vec{F}_{col}|_{\vec{V}=0} &= -c\int_{\theta=0}^{2\pi}d\theta g(\vec{x};t)\vec{\theta}_{x1}\\ &= -cR_B\pi\frac{\partial g(\vec{X};t)}{\partial\vec{X}}+O({R_B}^2)\\ &= +cR_B\pi\tilde{\beta}e^{-\tilde{\beta}\left(V_{cnt}(\vec{X};t)+V_{R0}(\vec{X})\right)}\\ &\quad\times\frac{\partial}{\partial\vec{X}}\left(V_{cnt}(\vec{X};t)+V_{R0}(\vec{X})\right)\\ &\quad+O({R_B}^2) \end{aligned}\quad (39)$$

となる．物理の教科書どおり，ロボットはポテンシャル $V(\vec{x};\vec{X};t)$ の負勾配に沿って動く．しかし (39) によれば，ロボットが感じているポテンシャル $V_{cnt}(\vec{x};t)+V_{R0}(\vec{x})$ は，物体をその正勾配に沿って動かしている．ロボット群は高いポテンシャル値 $V_{cnt}+V_{R0}$ を持つところから，そのより低い領域に流れ出す．そこで物体へのロボットの衝突力はポテンシャル値がより低い領域の方が大きくなるのである．このためロボット群は物体を，ポテンシャル値の低いところから高いところへと押し出すことになる．これが，なぜ (39) で衝突力 $\vec{F}$ の計算式にプラスが出てくるか，の物理的な理由である．

4　シミュレーション

4.1　数値計算

パラメータは 3 個あり対応する方程式は 2 成分をもつベクトル方程式 (30) と，温度 $\tilde{\beta}(t)^{-1}$ を決める時間の常微分方程式 (29) である．これらの 3 個の方程式いずれも $\vec{\alpha}(t)$ と $\tilde{\beta}(t)^{-1}$ が含まれるため，これらは同時に計算される．これらは非線形であり，ニュートン・ラプソン法や 2 分法といった典型的な方法で計算することができる．しかしこれらにはそれぞれの欠点がある．

- ニュートン・ラプソン法：言うまでもなく (30) の $\vec{F}$ や (29) の f の $\vec{\alpha}$ と $\tilde{\beta}^{-1}$ による偏微分が必要となる．関数が解析的にこれらのパラメータで表現されていれば数式により偏微分を書き下すことができるが，関数表現のプログラムが複雑になる．また本稿であれば $\vec{F}$ や f が $\vec{\alpha}$ で解析的に表現されていないため，数値微分が必要となりその精度への計算結果の吟味，という余分な作業が発生する．また計算初期値の設定および緩和係数 r，最大繰り返し回数 N_{max} あるいは許容誤差 $\varepsilon_{allowed}$ の調整が必要になる．
- 2 分法：この方法であれば偏微分は不要である．しかし，計算すべき変数を求める範囲の初期設定と N_{max} もしくは $\varepsilon_{allowed}$ の調整が必要になる．さらに 2 分法は本来は 1 変数に対する計算法なので，(30) や (29) により 3 変数を同時に計算するアルゴリズムが複雑になる．

以上の理由により本稿では 3 変数を，ある常微分方程式系の定常バランス値として計算した．初期値と任意時点での計算は，具体的には，それぞれは以下の通りである．

- 時間 $t=t_0$ での初期値計算：仮想時間 t' の下で，パラメータ $\vec{\alpha}(t')$ と $\tilde{\beta}(t')^{-1}$ が (30) の誤差により時間発展するとする．すなわち，仮想時間 t' についての 3 式の常微分方程式を次のように設定する．

$$\frac{d}{dt'}\vec{\alpha}(t') = -\vec{F}_{col}(\vec{X}(t_0),\vec{V}(t_0);\vec{\alpha}(t'),\tilde{\beta}(t')^{-1};t_0) +\frac{\partial}{\partial\vec{X}(t)}V_{B0}+\delta\vec{V}(t_0)+M\vec{A}(t_0) \quad (40)$$

$$\frac{d}{dt'}\tilde{\beta}(t')^{-1} = -f(\vec{X}(t_0),\vec{V}(t_0);\vec{\alpha}(t'),\tilde{\beta}(t')^{-1}) +\frac{E_{tot}-E_{obj}(t_0)}{N_R} \quad (41)$$

(40) で右辺にマイナスをつけるのは，ロボットの衝突による力 $\vec{F}_{col}$ が，ロボットに作用するポテンシャルの正勾配に比例し，$\vec{F}_{col}\sim+\frac{\partial V_{cnt}}{\partial\vec{x}}=\vec{\alpha}$ となるからである．このとき (40) は

$$\frac{d}{dt'}\vec{\alpha}(t')\sim-\vec{\alpha}(t') \quad (42)$$

となって $\alpha(t')$ の計算の収束性が確保できる．同様に (25) の第 1 項 $=\tilde{\beta}^{-1}$ であるので (41) の右辺のマイナス符号が $\tilde{\beta}(t')^{-1}$ の収束計算を保証する．

- 任意時点の計算：$\vec{\alpha}(t)$ については (40) の右辺で t_0 を t で置き換え，やはり仮想時間 t' の時間微分方程式を計算することで計算する．一方，$\tilde{\beta}(t)^{-1}$ についてはその $t=t_0$ での初期値が (41) から算出されているので，これを与条件とする時間常微分方程式 (29) の (仮想ではない実時間での) 時間積分により任意時点の $\tilde{\beta}(t)^{-1}$ が算出される．なお (28) において $\frac{\partial V_{cnt}}{\partial t}=\dot{\alpha}_1(t)\overline{x}_1+\dot{\alpha}_2(t)\overline{x}_2$ であり，ここで $\dot{\vec{\alpha}}(t)$ は時間差分で計算する，すなわち $\vec{\alpha}(t-dt)$ は 1 時点前の値として与えられており，さらに $\vec{\alpha}(t)$ は上記ですでに計算できているとして，$\dot{\vec{\alpha}}(t)=\frac{\vec{\alpha}(t)-\vec{\alpha}(t-dt)}{dt}$ である．また場合によっては $\vec{\alpha}(t)$ を $\tilde{\beta}(t)^{-1}$ と同時に計算する方法もある．

4.2 計算条件

計算パラメータは表 **2** に示すとおりである．辺長 2[m] の正方形領域の中で，中央に置いた一つの物体を，その東西南北四方のどこかにロボットを 1 台づつ，計 $N_0 = 4$ 台を配して出発する．ロボットと物体，ロボッ

meaning	parameter	value
geometry	$[-S_1, S_1]$	$S_1 = 1$[m]
	$\times[-S_2, S_2]$	$S_2 = 1$[m]
number of robots	N_0	4[−]
radius	robot: $\frac{a_R}{2}$	0.05[m]
	object: R_B	0.25[m]
mass	robot: m	0.3[kg]
	object: M	37.5[kg]
coefficient of restitution	e	1[−]
collision potential	σ_S	4[J]
	σ_v	4[J]
	n_S	12[−]
	n_v	12[−]
wall reaction	c_R	4[J]
	c_B	4[J]
	n_{cR}	12[−]
	n_{cB}	12[−]
friction coefficients	robot: γ	0[kg/s]
	object: δ	0[kg/s]

Table 2: Parameters

ト間，ロボットと壁および物体と壁の衝突は，いずれも Lennard-Jones で引力項のないソフトコア型のポテンシャルを取る．その比例定数とベキがそれぞれ σ_S，σ_v，c_R および c_B と，n_S，n_v，n_{cR} および n_{cB} である [2, 3]．計算時間刻みは，フィードフォワード計算のための連続体力学では計算が収束する $\Delta t = 1 \times 10^{-4}$[s]，また制御シミュレーションではエネルギーが保存する計算ケースでその保存性を確保する $\Delta t = 2 \times 10^{-4}$[s] とする．空間積分は (23) の I_x 計算で必要になるが，空間刻みは $2S_1 = 2S_2 = 2$[m] を 20 分割して数値的に定積分をする．

4.3 フィード・フォワード計算と PD 補償

目標 $\vec{X}^{req}(t)$ として X_1，X_2 ともに正側の壁 $S_1 = S_2 = +1$ に直線的に近づける軌道をとる．ロボットの初期配置は任意とし，配置に応じた系のエネルギーで決まるフィード・フォワード $\vec{\alpha}^{FF}(t)$ を計算しておく．本稿では $X_1^{req} = X_2^{req}$ と設定するので，対応して $\alpha_1^{FF} = \alpha_2^{FF}$ と計算される．制御されたトレンドは，次の項目に従いその良否を評価する．

- トラッキングに先立つ壁衝突すれば，制御は失敗．
- 指標 $PI \equiv \int_0^3 dt(\vec{X}^{req} - \vec{X})^2 < 2$ なら成功．
- 時間 $3 < t$ での壁衝突に起因する物体のジグザグ運動，また操作量のコストを無視．

初期配位 100 ケースでフィード・フォワードだけでの制御シミュレーションをすると，図 **6** に見るように，上述の条件で成功する回数がほぼ ~ 30[%] 程度となる．

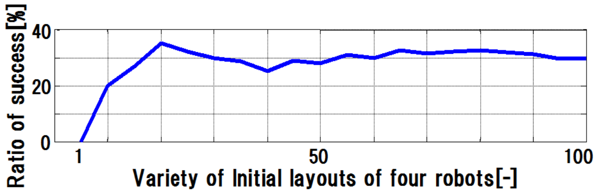


Fig. 6: Successful feedforward control in various initial latyouts of $N_0 = 4$ robots.

残りの ~ 70[%] の初期配位については，次に注意して **PD フィードバック補償**をする．

- 初期配位ごとに PD 定数を調整，
- フィード・フォワードの ~ 1[%] 程度の大きさ．

その結果 現状 PD 値でなお 5 ケースではフィードバック補償をしても制御が失敗している．

以上を図 **7** で **a)** にはフィード･フォワードのみ，**b)** は

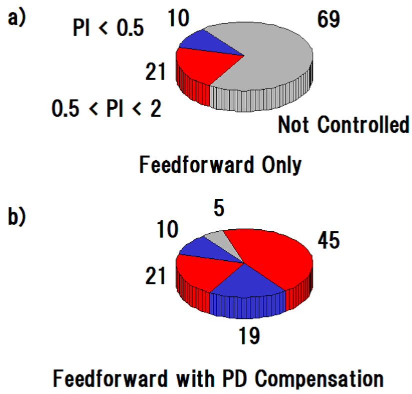


Fig. 7: Ratio of successful feedforward, a), and that with P+D feedback compensation.

フィードバック補償の結果をまとめ示す．指標 $PI < 0.5$ なら「成功」の中でも好成績と考えると，フィード・フォワードの 31 ケース中で 10 回，またフィードバック補償であれば 19 回が好成績を取っている．指標が最小 $PI = 0.130$ のフィード・フォワードによる物体の制御トレンドを図 **8** に，また操作入力と系の「温度」を図

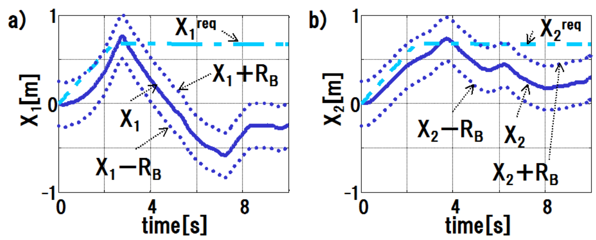


Fig. 8: Trends of the object in the minimum performance index.

9 に，それぞれ示す．注意点は，フィード・フォワード操作入力が $\alpha_1 = \alpha_2$ であり，また温度 $1/\tilde{\beta}$ はフィード・フォワード操作入力と同時に算出されるだけのもので制御には使わない，ということである．時間初期 $0 \sim t \sim 4$ に，ロボット群があたかも協調して物体を押し，トラッキングすることが，図 **10** で確認できる．ちなみに，この初期配位のエネルギーは 0.173[J] である．いっぽう最悪 (つまり最大) の指標値 ($PI = 4.384$)

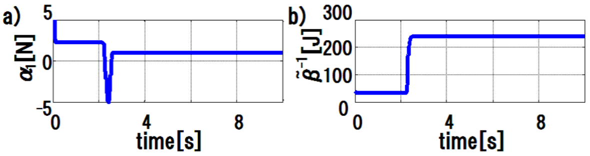


Fig. 9: Feedforward inputs and system “temperature” in the minimum performance index.

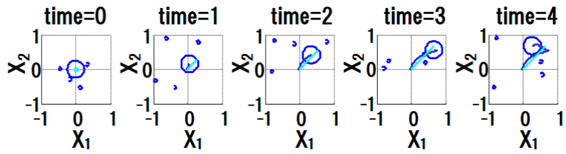


Fig. 10: Seeming cooperation of robots that makes the object track in the minimum performance index.

となる初期配位はエネルギー 0.557[J] であるが，これに関して，フィード・フォワードのみでの制御トレンドを図 **11** に，協調行動に失敗したかにみえるロボット

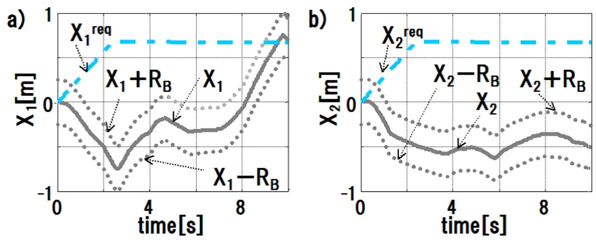


Fig. 11: Trends of the object in the worst performance index.

群と物体のレイアウトの時間推移を図 **12** に，それぞれ示す．レイアウト中 **b)** でフィード・フォワード操作入力の推移が時間の代表点についてバーチャートで確認できる．この操作入力は，図 9 のものとほぼ同一である．なお初期配位のエネルギーと制御成績の良否のあいだには，現状シミュレーションの限りでは，相関がない．さてこの最悪ケースをフィードバック補償すると図 **13** および図 **14** の結果となり，制御は成功する ($PI = 0.146$)．**b)**，**c)** にはフィードバック操作入力 α_1 と α_2 のそれぞれを示すが，図 12b) のフィード・フォワード値に比べ，その ~ 1[%] に満たない．

5　まとめと議論

少数台 (本稿では $N_0 = 4$ 台) のロボットが相互通信をすることもなく，また外界センサで外部環境を確認することもなく，一つの物体を閉じた平面領域でトラッキングすることができた．すなわち「意のままにモノを動かす」という「知的」挙動を，簡素なロボット少数台のシステムにより実現した．われわれはこれまでの研究 [2, 3] に従い，ロボットの「軌道描像」から「分布描像」に視点を移した．また本稿では，少数台のロボット群であってもエネルギーがボルツマン分布 (1) に従うことを確認できた．さらにポテンシャル V_{cnt} の数値計算を改良し，精度の高いフィード・フォワードを得た．これらにより，ロボット間の明示的な相互作用やロボットの学習メカニズムを入れることなく，ロボット群の協調行動を創発することが可能となった．

今後の課題は，ボルツマン分布 (1) の成立タイミングの精算，ロボット台数とトラッキング性能の相関の研究，フィードバック制御スキームの確立，さらにト

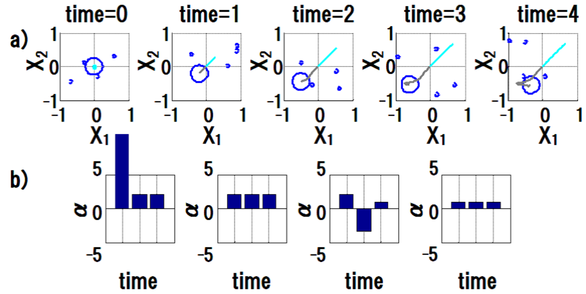


Fig. 12: Unsuccessful cooperation of robots in the worst performance index.

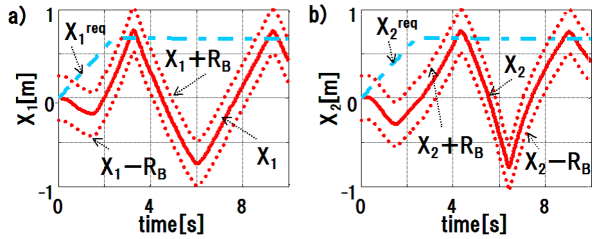


Fig. 13: Trends of the object in the worst performance index controlled under compensation of $P+D$ feedback.

イ・ロボット [6] を使った実験検証である．

参考文献

1) P.Hänggi and F.Marcheson, Artificial Brownian Motors: controlling transport on the nanoscale. *Review of Modern Physics*,**81**,383-442(2009).

2) Macroscopic Group Robots inspired By “Brownian Motion,” in *Handbook of Research on Design, Control, and Modeling of Swarm Robotics*, 2015(in press).

3) Transporting A Macroscopic Ob ect By Brownian Motion – An Ob ect As A Pollen Particle, Robots As Liquid Molecules –, in *Brownian Motion: Elements, Dynamics, and Applications*, 2015(in press).

4) 大沢文夫：大沢流手作り統計力学, 2012.

5) R.Pfeifer and J.Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence*, Bradford(2006).

6) http://www.elekit.co. p/product/4d522d39383032.

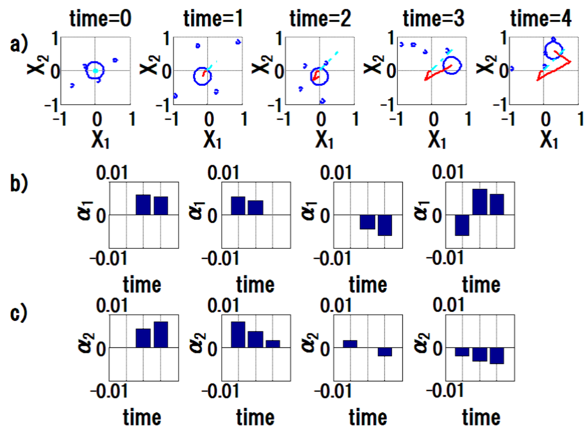


Fig. 14: Seeming cooperation of robots in the worst performance index controlled under compensation of $P+D$ feedback.

ふく射を考慮したアルミニウム板温度制御モデルに対する熱伝導率の推定

○細谷直紀　矢納陽　見浪護　松野隆幸　（岡山大学）

Estimation of T ermal onducti ity for Temperature ontrol Model of an Aluminum Plate it Radiati e Heat Transfer

N. osoya, A. Yanou, M. Minami and T. Matsuno (Okayama University)

Abstract− This paper considers the estimation of thermal conductivity for temperature control model of an aluminum plate with radiative heat transfer. In our research, we consider that the application of the control law which has been used in previous research to industrial eld. In the previous study, we were able to represent the model of interest in a formula, regardless of shape and material. In that case, it has been treated as already-known about the parameters. However, in the actual environment, physical parameters such as thermal conductivity change. So, we has been built an algorithm for estimating their parameters, but not the estimation results can be said to be appropriate value. Therefore, the model so far by considering the radiative heat transfer that was not taken into account, it was estimated parameters.

Key Words: ふく射, 熱伝導率, 推定則

1　緒言

温度制御は重化学工業において製品品質を決定する重要な要素の一つである．本研究では先行研究によって得られた制御則を金型などへ適用することで産業分野への応用を図ることを目指している．先行研究においてモデルの形状，材質を問わず制御対象を数式化することでモデリングすることが出来た．しかし，その際導出されたモデルは熱伝導率などの物理パラメータは定値として導出されている．ある媒体が熱を伝える際の物理量として熱伝導率が定義されており，この値が大きいほど熱が伝わりやすいことを表している．熱伝導率は温度とともに変化するため [2]，温度制御においては目標値変更や外乱の影響によって制御対象を構成する各部位の熱伝導率も変化し，結果として制御対象の特性も変化する．さらには制御対象の経年変化によって実際の熱伝導率が過去に取得したデータとは異なってしまい，結果として制御性能の劣化を招く可能性がある．したがって，制御性能を維持するためには制御対象のモデル構築とそれに基づき熱伝導率を推定できることが望ましい．これに対し著者らはアルミニウム板温度制御実験装置のモデルに対して熱伝導率の推定を行ったが，そこで推定された値は真値とされている値とはまったく異なる結果であった．そこで本研究ではこれまでのモデルの導出の際には考慮していなかったふく射を考慮することでモデルを実環境のものに近づけ，再度熱伝導率の推定を行った結果について報告する．具体的にはこれまでの研究で導出されたモデルに対しふく射によって熱が放出されること，熱の移動が行われることを数式化し，モデルに加えた上でパラメータ推定則を適用し，熱伝導率の推定を行う．

2　モデルの導出

アルミ板実験装置のモデルを Fig.1 に示す．モデルは薄いアルミの板一枚で構成されており，入力を行う部分と行わない部分の二つの部位に分けて考える．

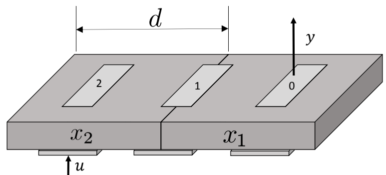


Fig. 1: Aluminum Plate Model

Fig.1 に示す各部位 x_1, x_2 の温度に関する状態量を $i = 1, 2$ に対して次のように定義する．

$$x_i = T_i \quad T_0 \tag{1}$$

ここで，T_i は各部位の温度，T_0 は室温を表す．
次に熱伝導に関する以下の式を用い，モデル化を行う．

$$E \ = \ \epsilon\sigma T^4 \tag{2}$$

式 (2) は放出に関する式で，ϵ は全放射率，$\sigma = 5.67$
$10 \ ^8$[W/m^{2}K^4] はステファン・ボルツマン定数，T[K] は物体の絶対温度，E[W/m^2] は放射する熱量 (放射能) を表している．

$$q = \quad _f(d\theta/dn) \tag{3}$$

式 (3) は熱伝導に関するフーリエの法則の式で，q は熱流速 $[W/m^2]$，　$_f$ は熱伝導率 $[W/mK]$，$d\theta/dn$ は熱流の温度傾斜 $[K/m]$ を表す．

$$q = \quad (\theta_s/\theta_f) \tag{4}$$

式 (4) は熱伝達とニュートンの冷却法則で，　は熱伝達率 $[W/m^2]$ を表す．

$$dQ = mc \quad d\theta \tag{5}$$

式 (5) は熱伝導に関する熱量と温度変化の関係式で，ここで dQ は熱量 $[J]$，c は比熱 $[J/kgK]$，m は質量 $[kg]$，$d\theta$ は温度変化 $[K]$ を表す．
以上をもとに，モデルに与えられる入力を u とおくと，Fig.1 の各部位 x_i の温度変化に関する式は次のように与えられる．なお，ここでは時間関数を表す添え字 (t) は省略して記述する．

$$mc\frac{dx_1}{dt} = \left\{ x_1 S_a + {}_f \frac{x_1 \quad x_2}{d} S_b \right\} \epsilon\sigma(x_1 + T_0)^4 S_a \tag{6}$$

$$mc\frac{dx_2}{dt} = \left\{ x_2 S_a + {}_f \frac{x_2 \quad x_1}{d} S_b \right\} \epsilon\sigma(x_2 + T_0)^4 S_a + u \tag{7}$$

m，S_a はそれぞれ部位 x_1, x_2 の質量，外気との接触総表面積を表し，S_b，d は部位 x_1，x_2 の接触面積とそれぞれの部位の横幅を表す．

また，アルミ板温度制御実験装置のパラメータを Table 1 に示す．

Table 1: Aluminum Plate Model Parameters

Density of aluminum	ρ	2700[kg/m^3]
Specific heat of aluminum	c	917[J/kgK]
Heat transfer coefficient		20[W/m^{2}k]
Thermal conductivity	$_f$	238[W/mK]
Width of plate	d_1	250[mm]
Thickness of plate	d_2	10[mm]
Length of plate	d_3	120[mm]
Initial Temperature	T_1,T_2,T_0	300[K]
Emissivity	ϵ	0.07

3 実機実験との比較

前章で得られた式 (6),(7) が妥当であるか否かの検証を行うために実機実験とシミュレーションの比較を行った．共に一定入力である 10[W] を与え続けた際の出力を確認した．なお実機の写真を以下の Fig.2 に示す．

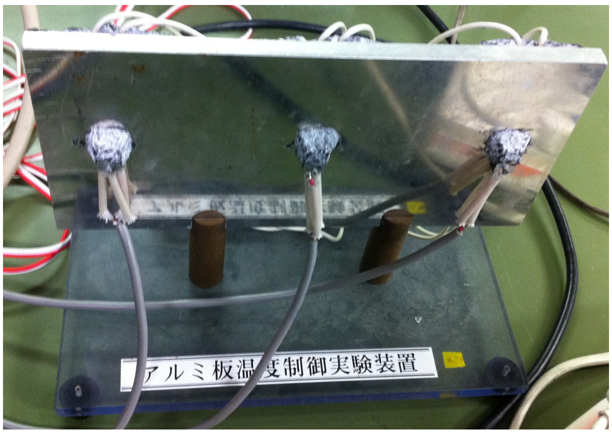


Fig. 2: Aluminum Plate Temperature Control Experimental Device

以下の Fig.3 にふく射を考慮していないモデルに対するシミュレーション結果を，Fig.4 に式 (6),(7) で与えられるふく射を考慮した際のモデルに対するシミュレーション結果を，Fig.5 に実機実験の結果を示す．

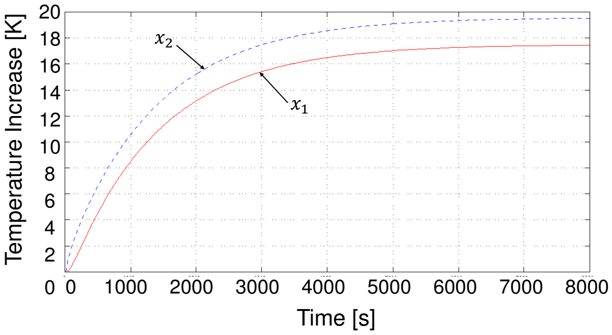


Fig. 3: Simulation Result(Not Considering Radiation)

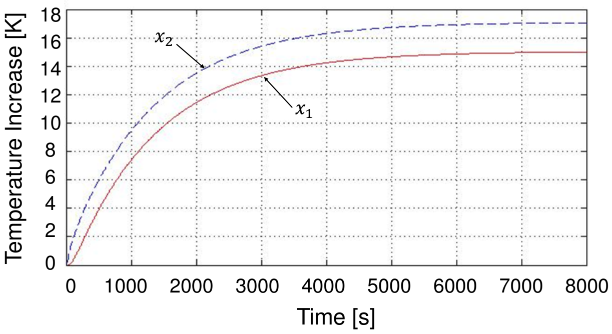


Fig. 4: Simulation Result(Considering Radiation)

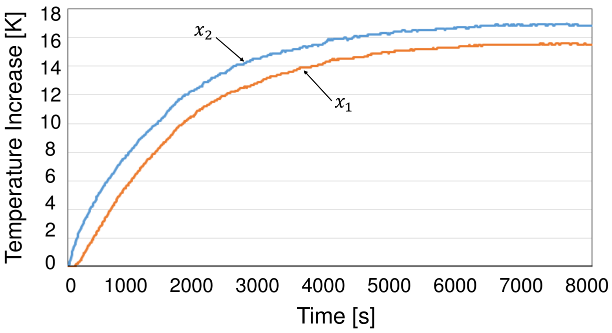


Fig. 5: Experimental Result

以上の結果よりふく射を考慮していないモデルでは実機実験の結果に比べて温度が上がりすぎていることが分かり，ふく射を考慮したモデルを用いることではぼ実機実験の結果と一致していることが分かる．このことより今回導出したモデルはある程度妥当であるといえる．

4　熱伝導率の推定

熱伝導率の推定を拡張カルマンフィルタによって行うため，モデルの式 (6)，(7) を実験データを取得した際のサンプリング時間 Δt を用いて以下のように離散化する．なお以下では時刻 $t = k\Delta t(k = 0, 1, 2, \quad)$ における状態量 $x_i(t)$ を $x_i(k)$ で表すものとする．

$$\frac{dx_i(t)}{dt} = \frac{x_i(t+\Delta t) \quad x_i(t)}{\Delta t}$$

熱伝導率 $\;_f(k)$ を含む新しい状態量 $z(k)$ を定義する．

$$\mathbf{z}(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \;_f(k) \end{bmatrix}$$

すると次式の拡大系を得ることが出来る．

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{f}(\mathbf{z}(k)) + \mathbf{b_{ue}}u(k) && (8) \\ y(k) &= h(\mathbf{z}(k)) && (9) \end{aligned}$$

ここで

$$\begin{aligned} \mathbf{f}(\mathbf{z}(k)) &= \begin{bmatrix} f_1(\mathbf{z}(k)) \\ f_2(\mathbf{z}(k)) \\ f_3(\mathbf{z}(k)) \end{bmatrix} \\ \mathbf{b_{ue}} &= \begin{bmatrix} 0 \\ \frac{\Delta t}{mc} \\ 0 \end{bmatrix} \\ h(\mathbf{z}(k)) &= z_1(k) \end{aligned}$$

である．さらに $f_i(\mathbf{z}(k))(i = 1, 2, 3)$ は以下で与えられる．

$$\begin{aligned} f_1(\mathbf{z}(k)) &= \left\{1 \quad \frac{\Delta t}{mc}(\quad S_a + z_3(k)\frac{S_b}{d_1})\right\} z_1(k)) \\ &\quad + \frac{\Delta t}{mc}\left(\frac{z_3(k)S_b}{d_1}\right) z_2(k) \\ &\quad \frac{\Delta t}{mc}\epsilon\sigma S_a(z_1(k)+T_0)^4 && (10) \\ f_2(\mathbf{z}(k)) &= \left\{1 \quad \frac{\Delta t}{mc}(\quad S_a + z_3(k)\frac{S_b}{d_1})\right\} z_2(k) \\ &\quad + \frac{\Delta t}{mc}\left(\frac{z_3(k)S_b}{d_1}\right) z_1(k) \\ &\quad \frac{\Delta t}{mc}\epsilon\sigma S_a(z_2(k)+T_0)^4 && (11) \\ f_2(\mathbf{z}(k)) &= z_3(k) && (12) \end{aligned}$$

つぎに，式 (8)，(9) に対して拡張カルマンフィルタを適用する．まず状態推定値の初期値 $\hat{\mathbf{z}}(0)$ を，その平均が $\mathbf{z_0}$，分散が $\mathbf{\Sigma_0}$ として

$$\begin{aligned} \hat{\mathbf{z}}(0) &= \mathbf{z_0} && (13) \\ \mathbf{P}(0) &= \mathbf{\Sigma_0} && (14) \end{aligned}$$

とおく．またシステム雑音の分散が σ_v^2，観測雑音の分散が σ_w^2 であるとする．さらに $k = 1, 2, \quad , N$ に対して次のように事前状態推定値を計算する．

$$\hat{\mathbf{z}} \quad (k) = \mathbf{f}(\hat{\mathbf{z}}(k \quad 1)) + \mathbf{b_{ue}}u(k \quad 1) \qquad (15)$$

次に式 (8)，(9) を利用して以下のヤコビアンを計算する．

$$\begin{aligned} \mathbf{A}(k \quad 1) &= \left.\frac{\partial \mathbf{f}(\mathbf{z}(k))}{\partial \mathbf{z}(k)}\right|_{\mathbf{z}(k)=\hat{\mathbf{z}}(k \quad 1)} && (16) \\ \mathbf{c}^T(k) &= \left.\frac{\partial h(\mathbf{z}(k))}{\partial \mathbf{z}(k)}\right|_{\mathbf{z}(k)=\hat{\mathbf{z}}^-(k)} && (17) \end{aligned}$$

本論分ではシステム雑音が入力に直接加わるものとし，式 (16) を利用して次のように事前誤差共分散行列 $\mathbf{P} \quad (k)$ を計算する．

$$\mathbf{P} \quad (k) = \mathbf{A}(k \quad 1)\mathbf{P}(k \quad 1)\mathbf{A}^T(k \quad 1) + \sigma_v^2\mathbf{b_{ue}}\mathbf{b_{ue}}^T \qquad (18)$$

式 (17)，(18) を利用してカルマンゲイン $\mathbf{g}(k)$ を計算する．

$$\mathbf{g}(k) = \frac{\mathbf{P} \quad (k)\mathbf{c}(k)}{\mathbf{c}^T(k)\mathbf{P} \quad (k)\mathbf{c}(k) + \sigma_w^2} \qquad (19)$$

以上より状態推定値 $\hat{\mathbf{z}}(k)$，事後誤差共分散行列 $\mathbf{P}(k)$ を求めることが出来る．

$$\begin{aligned} \hat{\mathbf{z}}(k) &= \hat{\mathbf{z}} \quad (k) + \mathbf{g}(k)\left\{y(k) \quad h(\hat{\mathbf{z}} \quad (k))\right\} && (20) \\ \mathbf{P}(k) &= \left\{\mathbf{I} \quad \mathbf{g}(k)\mathbf{c}^T(k)\right\}\mathbf{P} \quad (k) && (21) \end{aligned}$$

これらの計算手順を繰り返し行うことによって，モデルの温度と熱伝導率の推定値が得られる．

5　推定結果

取得した実験データは Fig.5 で得られたものを用い，そのサンプル数は 800 である．サンプリングタイムは $\Delta t = 10$ となっている．また，システム雑音の平均を 0，分散を 0.01^2，観測雑音の平均を 0，分散を 0.001^2 とおいた．拡張カルマンフィルタの誤差共分散行列の初期値は $\mathbf{P}(0) = 100\mathbf{I}$，推定する熱伝導率の初期はアルミニウムの熱伝導率とされている 238 を与えた．その他モデルのパラメータはシミュレーションで用いた Table.1 の値を用いている．

以上の条件の下，シミュレーションによって得られたデータを用いてモデルの部位 x_1，x_2 および熱伝導率 $\;_f$ を推定した結果をそれぞれ Fig.6，Fig.7 及び Fig.8 に示す．いずれのグラフにおいても実線が推定値，破線が部位 x_1 に対する測定温度データを表している．Fig.6 に x_1 の上昇温度，Fig.7 に x_2 の上昇温度，Fig.8 に熱伝導率の推定結果を示している．この三つのグラフより作成した推定則及びプログラムが正しい挙動をしていることが確認できる．

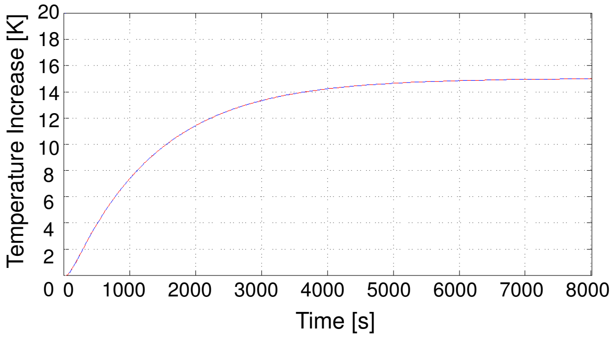


Fig. 6: Estimation Result by Simulation(x_1)

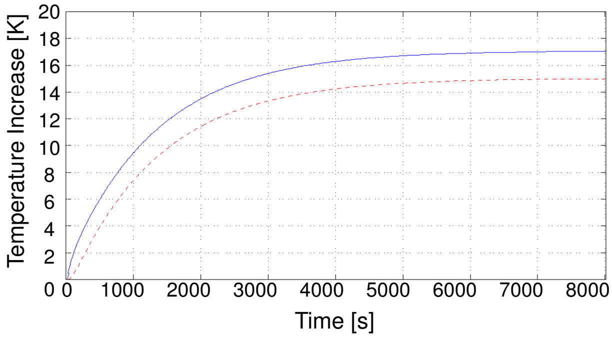


Fig. 7: Estimation Result by Simulation(x_2)

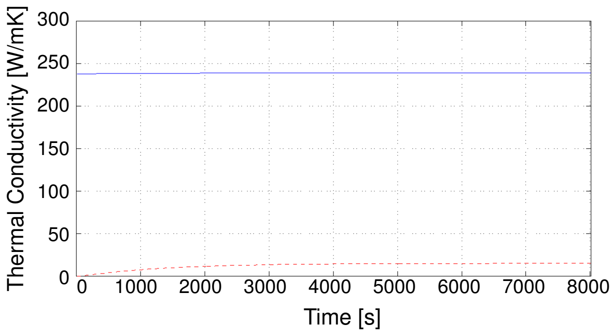


Fig. 8: Estimation Result by Simulation($_f$)

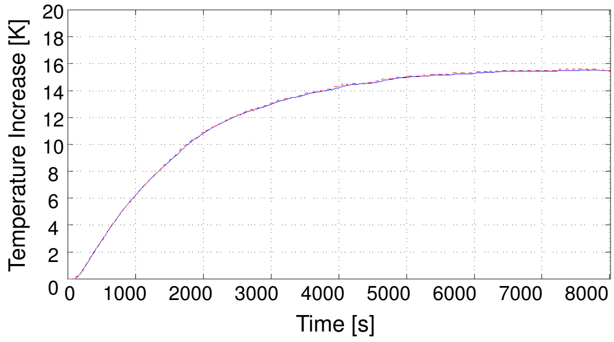


Fig. 9: Estimation Result by Experiment(x_1)

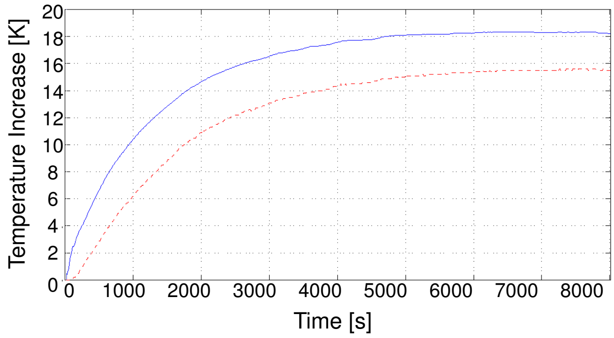


Fig. 10: Estimation Result by Experiment(x_2)

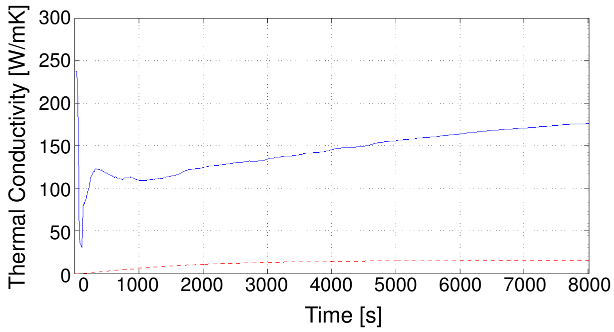


Fig. 11: Estimation Result by Experiment($_f$)

また，同様の条件の下，モデルの部位 x_1，x_2 および熱伝導率 $_f$ を実験結果より得られたデータを用いて推定した結果をそれぞれ Fig.9，Fig.10 及び Fig.11 に示す．いずれのグラフにおいても実線が推定値，破線が部位 x_1 に対する測定温度データを表している．Fig.9 には実験データに基づいて推定された部位 x_1 の上昇温度が示されている．実験データとモデルにより推測された値が一致していることが分かる．Fig.10 には実験データに基づいて推定された部位 x_2 の上昇温度が示されている．Fig.5 と比べて差異があるが，これはモデル化誤差が現れているものと推測できる．Fig.11 には実験データ基づいて推定された熱伝達率の値を示している．今回行った実験は 800 のデータを用いており実験時間にして約二時間半であるが，現状データの数が足りないことが分かる．しかしアルミニウムの熱伝導率とされている 238 という値に近づいていっていることが分かる．

6　結言

本報告ではアルミ板温度制御実験装置に対するふく射を考慮したモデルの構築を行った上でその離散時間モデルの拡大系を構築し，実機実験とシミュレーションの比較を行うことでモデルの妥当性の検証を行った．さらに，実験データを拡張カルマンフィルタに適用することで，構築したモデルの各部位の上昇温度と熱伝導率の推定結果を示した．これにより，未知の物性値に対してオフラインで推定値を得ることが出来た．今後の課題としてデータ数を増やすことにより熱伝導率が収束していくことを確認すると共にその値が妥当であるかどうかの検証を行う必要がある．また熱伝導率以外の熱伝達率や比熱などの各パラメータの同時推定，及びオンライン推定が挙げられる．

参考文献

1) 矢納陽，内田茂樹，細谷直紀，見浪護，松野隆幸，アルミ板温度制御モデルに対する熱伝導率の推定，第 57 回自動制御連合講演会，333/337(2014)

2) 京セラ株式会社ファインセラミック事業本部，Characteristics of Kyocera Technical Ceramics, 京セラ株式会

社 (2014)

3) 日本機械学会，演習　伝熱工学，53/63，丸善出版 (2013)

4) 足立修一，丸田一郎，カルマンフィルタの基礎，東京電機大学出版局 (2012)

5) 矢納陽，内田茂樹，細谷直紀，見浪護，松野隆幸，耐火断熱れんがの熱伝導率の推定，第 58 回自動制御連合講演会 (2015)