

モデル予測制御向けニューラルネットワークの構成検討

○堀口辰也 広津鉄平 (日立製作所)

Study of Neural Network Structure for Model Predictive Control

* T. Horiguchi and T. Hirotsu (Hitachi, Ltd.)

Abstract — To achieve both improvement of accuracy and acceleration of calculation for Model Predictive Control(MPC), modeling method using neural networks(NNs) is proposed. Proposed NN enables time-independent prediction, which resolves both accumulation of prediction error and restriction of forward prediction caused by sequentiality of modeling. Experimental result show its effect in modeling of nonlinear controlled objects.

Key Words: Model Predictive Control, Neural Network, Plant Modeling

1 はじめに

近年、画像処理や音声認識分野をはじめとする様々な分野で、機械学習、深層学習技術が注目を集めている。画像認識処理のベンチマーク¹⁾にて深層学習が上位を席巻した後、音声認識技術の精度向上²⁾、囲碁を例題とする対人ゲームでの勝利³⁾と続いているが、これに限らず近年では産業用ロボットの制御⁴⁾や自動運転制御⁵⁾の技術開発も進められており、産業分野について同技術のさらなる適用進展が考えられる。このような状況に対し、本稿では深層ニューラルネットワーク(Deep Neural Network: DNN)の制御適用例として、モデル予測制御を想定し、同制御に適したDNN構成を提案するものである。

2 モデル予測制御

モデル予測制御は最適制御と呼ばれる制御の一種で、制御対象に対し、同制御対象のモデルを用いることで一定期間将来までの制御対象の状態最適化を行う、モデルベース制御の一種である。最適制御のうち、特に制御対象の状態をリアルタイムにフィードバックするものをモデル予測制御と呼ぶ⁶⁾。古くは石油化学プラントなどの大規模なプラントにおける制御最適化⁷⁾のような、多入力・多出力を特徴とする大規模なシステムの制御に用いられてきた制御である。

そのベースは数理計画問題(最適化問題)で、数式(1)に示す制御対象の動作を表現するモデル(本例では非線形モデルを仮定する)

$$\dot{x} = f(x, u, t) \quad (1)$$

および、制御対象毎に固有な制約条件Cのもとで、下記のユーザにより定義される評価関数Jの最適化(多くは最大化もしくは最小化)を図る問題と定義される。

$$J = \phi(x(T), u(T), t(T)) + \int_t^{t+T} L(x(t+\tau), u(t+\tau), \tau) d\tau \quad (2)$$

ここで、 x は状態量(制御対象が持つ物理量)、 u は操作量(制御対象へ入力される物理量)を表すとする。

モデル予測制御、特に非線形な制御対象を扱う非線形モデル予測制御には、大別して次の課題がある。

1) 処理量、およびそれに伴う制御演算時間

先述のように、モデル予測制御は制御対象の将来状態に及ぶ状態最適化を行うことから、従来用いられる制御演算に対し演算量が大きくなる。そのため、先述の大規模なシステムの制御例⁷⁾のように比較的制御周期が長い(数 sec 以上)ものや、もしくは高い計算性能が利用可能な分野に限られてきた。一方で近年では、ロボット制御⁸⁾や自動運転⁹⁾、等への制御適用を目差し、高速に動作するアクチュエータ(数 msec~数十 msec)への適用に向けた演算の高速化が求められている。

2) モデリングの精度

モデル予測制御は、前述のようにコントローラが保持する制御対象モデルを用いた予測を基に、最適化を行うモデルベース制御であり、その精度は制御性能に直接影響することから、高精度なモデリングが求められる。特に大規模なシステムや複雑な構造のアクチュエータ等、非線形性の強い制御対象のモデリングも重要となっている。

3 従来方式と課題

先章にて説明したモデル予測制御は数理計画問題に帰着される。その解法は数値的解法、探索的解法に大別されるが、ともに制御対象のモデルを用いて最適解(操作量)の導出を行うものである点で共通している。本章ではモデル予測制御の各解法の特徴と課題、および従来の制御対象モデリング方式の課題を説明する。

3.1 従来解法の方式と課題

1) 数値的解法

数式(2)に定めた評価関数の勾配情報に基づき、最適解を導出する解法である。単峰性問題に対して最適解の導出を保証できること、またC/GMRES法¹⁰⁾のような高速解法が提案されていることが特徴である一方、

評価関数が不連続な値を取るケースや多峰性問題への対応が困難である。

2) 探索的解法

複数の解候補値(探索点)を生成し、それらについて、評価関数を用いた評価、およびアルゴリズム毎に定められるルールに従う更新を繰り返すことで、最適解を探索する手法である。解の精度が探索点数および解候補の配置に依存するが、評価関数の勾配情報を用いないため、不連続な評価関数構成や、多峰性問題へ対応が可能となる。一方で、実時間内での最適解の導出は保証されないこと、また各探索点について、制御対象モデルを用いた挙動予測と評価を行う必要があり、演算負荷が増大することが課題となる。

3.2 従来モデリング手法の課題

上記の演算を行うにあたり、先述のような制御対象のモデルが必要となる。一般的なモデリング形式は先述の数式(1)に基づき時間方向に逐次的な状態更新を行うもので、Fig. 1のように構成・演算される。

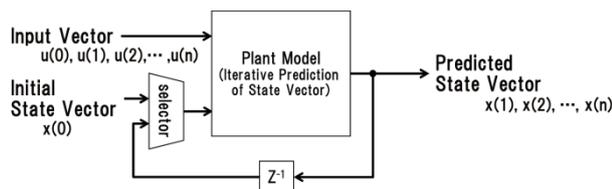


Fig. 1: Target Actuator for Modeling

このようなモデリング方式には、次の課題がある。

1) 制御対象モデリング自体の難しさ

モデル予測制御が対象とする、大規模なシステムや近年の複雑化するアクチュエータは、内部の各状態量が相互作用する連成モデルで記述されることが多い。そのため、数式(1)のような単純な因果関係モデルを用いて精度よくモデリングを行うことが困難なケースがあり、モデリング誤差の増大もしくはモデル予測制御における計算コストの増大が懸念される。

2) 蓄積する演算誤差

制御対象モデルを用いた演算には、上記1.の制御対象のモデリング誤差に加え、同モデルの積分演算により状態量予測値を得ることによる積分誤差が存在する。この両誤差が、モデル予測制御に必要な時刻分だけ繰り返し演算され蓄積することから、予測過程での精度低下が課題となる。

4 提案方式

4.1 ニューラルネットワークモデル予測制御の構成

本稿では、前述のモデル予測制御の解法のうち、制

御対象および評価関数の複雑さから多峰性問題に対応できる探索的解法を選択し、同解法に適したニューラルネットワークを構築した。なお、探索的解法のアルゴリズムには、人工蜂コロニーアルゴリズム¹¹⁾を採用している。3.1節にて説明したように、人工蜂コロニーアルゴリズムについても多数の探索点を用いることによる演算負荷の課題が残るが、近年、大規模な並列演算に適するGPU(graphic Processing Unit)やFPGA(Field-Programmable Gate Array)の利用が産業分野にも広がりつつあることから、演算負荷についてはこれらのデバイスを用いることにより解消可能と考えている。また、本稿で提案するDNNを用いた制御対象モデリングについて、モデル予測制御の演算内にDNNを用いることを想定している点も、同様の理由によるものである。

4.2 モデリング対象

本稿におけるモデリング例として、Mathworks社より提供されている油圧ソレノイドモデル¹²⁾(MATLAB/Simulinkモデル)を選択した。同シミュレーションモデルを用いた理由としては、同油圧ソレノイドが、実機環境において複数の物理ドメインにより構成される連成モデルであるためであり、3.2節で述べた現行の制御対象モデリングにおける課題を適切に表すと考えられるためである。本検討にて利用したモデルの概略図を、Fig. 2に示す。

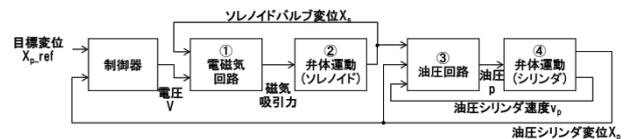


Fig. 2: Modeling of Target Actuator

同モデルにおいては、ソレノイドバルブ(図中①、②)に印加する操作量(電圧 v)により、ソレノイドバルブの稼働子変位 X_s を変化させることで、油圧回路③への油圧供給量を調整し、油圧シリンダ④の弁体位置 X_p を目標値に追従させる制御が行われる。同モデルにおける状態量 x は、上記のソレノイド弁・油圧シリンダ変位値 X_s, X_p に加え、両変位についての速度 v_s, v_p および加速度 a_s, a_p 、ほか、ソレノイドバルブ内の状態量としての電流値 I 、磁束 ϕ 、油圧回路内の状態量としての油圧 p から構成される。詳細については⁷⁾を参照されたい。本検討においては、このような油圧ソレノイドバルブシステムに対し、モデル予測制御での評価関数の構成を意識し、NNを用いたモデリングを行った。

4.3 一括出力型 NN モデリング方式の提案

従来の制御対象のモデリングでは、3.2節に示すように制御対象モデルを用いた逐次的な状態更新がとられるが、先述のように次状態予測を繰り返すため、予測過程での誤差が蓄積し精度が低下する問題があった。また上記の予測方式では、次状態の予測のために制御対象に含まれる状態量 x の全要素について演算が行わ

れるが、一方でモデル予測制御においては必ずしも前述の全ての内部状態が制御対象に加える操作量の評価に用いられるわけではない。まず、本評価対象である油圧シリンダ制御システムの評価関数構成を、以下に示す。

$$\varphi(x(T), u(T), t(T)) = (tgt - X_p)^2 + I(T)^2 + P(X_s) \quad (3)$$

ただし、 $P(x)$ は下記数式(4)で示されるペナルティ関数で、次のような形状をしているとする。

$$P(X_s) = \begin{cases} Penalty & (X_s < 0 \parallel X_s > Limit) \\ 0 & (others) \end{cases} \quad (4)$$

$$L(x(T), u(T), t(T)) = (tgt - X_p)^2 + I(T)^2 + P(X_s) \quad (5)$$

上記のような評価関数を構成する意図は、下記の通りである。

- 第1項(シリンダ変位 X_p に関する) : システムの制御目的(目標位置追従)
- 第2項(電流値 I に関する) : システム全体として低消費電力を指向
- 第3項(稼働子変位 X_s に関する) : 電磁弁の動作範囲上下端への衝突による装置摩耗を防止

評価関数の構成から分かるように、上記3点以外の状態量は、評価対象とする状態量の導出のみを目的に演算されることになる。言い換えれば、制御に必要な状態量のみを予測できれば、それ以外の状態量の導出は不要となる。本稿で提案する DNN を用いたモデリング方式は、このような予測制御の特徴も合致する。

Fig. 3 に本章で提案する一括予測型 NN の構成を示す。本図における操作量 u は電圧 V を表すが、DNN の特徴である、入出力関係の近似機能に着目することで、状態量 x は 4.2 節で説明した従来のモデリング方式とは異なり、電流値 I 、稼働子変位 X_s およびシリンダ変位 X_p の 3 状態量のみで構成することが可能となる。

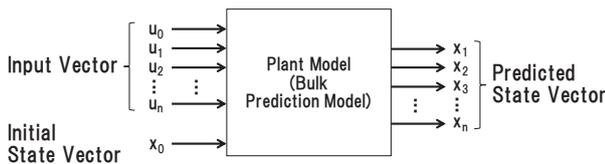


Fig. 3: Structure of Proposed NN Model

提案する一括予測型 NN 方式は、制御対象に印加する操作量 $u(0) \sim u(n)$ と、モデル予測制御の評価関数演算に必要な状態量初期値 $x(0)$ を一括してモデルに入力し、モデルから制御対象の挙動変化予測 $x(1) \sim x(n)$ を一括して出力するため、逐次的な予測値の更新が不要となる点で、従来方式(逐次更新型と呼ぶ)と大きく異なるものである。このようなモデル構成により、下記3点のメリットが得られる。

1) 制御対象の挙動予測演算の短縮

Fig. 1 に示す従来方式では、予測期間の増大に比例して制御対象モデルの計算回数が増大する。本方式においてはモデルが大規模化するものの予測値を一度の

演算で出力することが可能となる。従来構成の小さなモデルは、1 度のモデル演算においては計算速度で優位と考えられるが、同じモデルを予測期間分だけ繰り返す方式であるため、本方式での予測更新演算時間の短縮が期待される。これにより、3.1 節で述べたモデル予測制御の課題である、制御演算に要する時間を解消することが期待される。

2) 予測誤差蓄積の防止

前述の逐次更新型の予測では、予測値を基に予測を繰り返す構造上、モデルに予測誤差が発生していた場合には予測誤差が蓄積し、前章で示したように、予測が長期になるにつれ、予測精度が低減する。一方で本方式では、一括して出力を演算することから、このような誤差の蓄積を回避可能である。これにより、3.2 節で述べた従来モデリング方式による誤差蓄積が解消され、モデル予測制御による制御精度の改善が期待される。

3) 実動作データに基づいたモデリング

センサにより取得される実動作データの近似によるモデリングが可能となり、モデリングの複雑さを解消することができる。これにより、3.2 節で述べた制御対象モデリングの難しさを、入出力データの取得で解消することができる。

対するデメリットは、モデル入出力数が予測期間に応じて増大することによる、モデル規模の増加である。NN モデルが表現すべき対象(入出力関係)が大規模化するため、NN の規模が増大する。これは実装に必要な回路面積(=コスト)の増大に繋がる。

4.4 一括出力型 NN を用いた制御対象モデリング

4.2 節にて説明した油圧ソレノイドモデルを用い、4.3 節にて説明した一括出力型 NN によりモデリングを行う方法について説明する。これには Fig. 2 に示した制御対象のモデルから得られる制御対象の状態量データ (Tab. 1) を基に学習データを作成する必要がある。

4.3 節で述べたように提案する NN は、モデル予測制御において制御対象の挙動を予測するための、赤線で囲まれた状態量初期値 $x(0)$ と、時系列の操作量 $u(0) \sim u(n)$ とを一括で入力とし、青線で囲まれた制御対象の挙動変化予測 $x(1) \sim x(n)$ を一括で出力するものである。なお、赤点線で囲った部分は、制御対象の過去のデータであり、必要に応じて NN への入力として追加することも可能である。

実機を用いる際にも同様に、制御対象への入力データ(操作量)と、制御対象に登載するセンサデータ(操作量)を用い、学習データを作成することが可能である。

Tab. 1: Learning Data Creation for Proposed NN

Timestep	Control input	State Value			
		I	X_s	X_p	
t	V	I	X_s	X_p	
\vdots					
1763	10.318	5.2325	0.762	19.474	(optional) Input Data
1764	10.332	5.2395	0.762	19.471	
1765	10.304	5.2255	0.762	19.476	Input Data
1766	10.29	5.2185	0.762	19.479	
1767	10.276	5.2115	0.762	19.482	Output Data
\vdots					
1794	9.898	5.0225	0.762	19.552	

本稿においては、モデル予測制御において 30 制御周期分の挙動予測を行うとし、提案する一括出力型 NN を構成した。そのため、入力数は 33 入力(操作量 $u * 30$ と状態量 $3 * 1$)、出力数は 90(状態量 $3 * 30$)となる。実装にあたり 8 層の MLP を用い、各層のニューロン数は 50(各層で共通)とした。また、学習条件としては、最適化手法に Adam を、活性化関数に LReLU(Leaky Rectified Linear)を用いた。

4.5 評価結果

Fig. 4 に、制御周期 30 周期分の予測を行うモデル予測制御を例に、本稿にて提案する一括出力型 NN を用いた制御対象挙動予測の結果を示す。上記の結果は、中間層 7 層、各層のニューロン数 200、学習回数 1000 回、事前学習有りの条件で実装を行ったもので、青線が学習データを、赤線が各時刻を開始とする制御対象の 30 周期分の挙動予測(NN の出力)を表している。また、NN の各入出力の値はそれぞれ正規化されている。一括出力型 NN より得られる、出力(電流値、ソレノイド弁変位および油圧シリンダ弁変位)全体での二乗平均誤差は、動作状態の変化点において大となる傾向があるが、最大で 0.04459 に留まった。主に動作状態の変化点において、誤差特性が悪化していることが分かるが、提案する一括出力型 NN の構成により、逐次的な挙動予測演算が不要となるため、動作状態の変化点における誤差の大きい予測値が、以降の予測値に影響を及ぼさず、誤差の増大・発散が防止できていることが分かる。

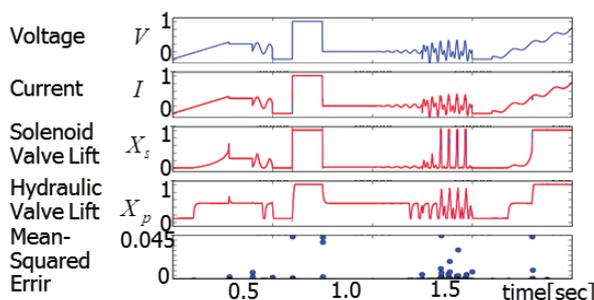


Fig. 4: Modeling Result with Proposed NN

5 おわりに

本稿では、深層学習技術の制御適用の一例として、モデル予測制御の特徴に着目した一括更新型 NN を提案した。同構成により、制御対象の実動作データに基づく制御対象モデル構築が可能となる。また、モデル予測制御の課題であった、制御対象挙動予測の精度向上、および演算時間の短縮が期待される。

今後、本稿にて着目したモデリング方式を実制御に組み込み、制御精度および演算速度について評価を行う予定である。また、一括更新型 NN の課題である NN の大規模化について、省面積実装方式の開発を今後行う予定である。

参考文献

- 1) “ILSVRC2012” : <http://image-net.org/challenges/LSVRC/2012/>
- 2) D. Amodei, R. Anubhai, et al., : Deep Speech 2: End-to-End Speech Recognition in English and Mandarin, arXiv:1512.02595, 2015.
- 3) D. Silver, A. Huang, C.J. Maddison et al., : Mastering te game of Go with deep neural networks and tree search, Nature, vol.529, pp.484-489, 2016.
- 4) Fanuc 株式会社 : <http://www.fanuc.co.jp/ja/product/show/2015/201512irex.html>
- 5) トヨタ自動車株式会社 : <http://newsroom.toyota.co.jp/jp/detail/10680141/>
- 6) 大塚敏之 : 非線形最適制御入門, コロナ社, 2011.
- 7) D.M.Prett, M.Morari: The Shell Process Control Workshop, Process Control Research: Industrial and Academic Perspectives, Butter-worths, Boston, 1837.
- 8) Z. F. G. Wei : Model Predictive Control for Industrial Robots, Robotics 2000, pp. 263-269, 2000.
- 9) 西羅光, 川邊武俊 : 自動車の最適経路生成, 計測と制御 Vol. 45, No. 3, pp 209-215 , 2006
- 10) T. Ohtsuka : A Continuation/GMRES Method for Fast Computation of Nonlinear Receding Horizon Control, Automatica, Vol. 40, No. 4, pp. 563-574, 2004
- 11) D. D. Karaboga : An Idea Based On Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes Univ. Engineering Faculty, Computer Engineering Department, 2005.
- 12) Mathworks : PWM 駆動油圧サーボ機構のモデル化, http://jp.mathworks.com/help/stateflow/examples/modeling-a-pwm-driven-hydraulic-servomechanism_ja_JP.html

自己組織的ターゲット検出モデルによる 分布推定アルゴリズムについて

○岩崎悟 畠中利治 (大阪大学)

Estimation of distribution algorithm based on self-organizing target-detection model

*S. Iwasaki and T. Hatanaka (Osaka University)

Abstract— Evolutionary computation is one of the promising methodologies to provide an actual solution for various function optimization problems. Almost evolutionary computation models have heuristic procedures in its updating search points or probability density functions. In this study, in order to give theoretical aspects of estimation of distribution algorithms (EDAs), we propose an EDA that updates probability density functions in accordance with a well-studied partial differential equation. Then, we show numerically some typical behaviors of the the proposed EDA with tuning a parameter in the partial differential equation.

Key Words: Estimation of distribution algorithms, Partial differential equation, Parameter tuning

1 はじめに

関数最適化問題は、 D 次元の実数ベクトルを説明変数とし、目的関数 $f(\vec{x})$ の最大または最小値を求める問題である。特に、 $f(\vec{x})$ の勾配、単峰性か多峰性かといった景観や変数分離の可否などの情報が事前に得られず、観測点 \vec{x} における関数値 $f(\vec{x})$ のみを知り得るという場合をブラックボックス関数最適化といい、現実の多くの問題はこのクラスとしてモデル化することができる。

遺伝的アルゴリズム、進化戦略、群知能などによる関数最適化法は、複数の探索点を用いた確率的な逐次探索法であり、目的関数の評価値のみの情報を用いて解探索を進めることができることから、ブラックボックス関数最適化に対する有用なツールとみなされ、近年では、さまざまな分野での応用も研究されてきている¹⁾。しかし一方で、大域的最適解への収束の保証がないことや、効率よく探索を進めるためのパラメータ設定に実験的な試行錯誤もしくは知識が必要になるなど、理論的側面に課題がある。

この原因として、例えば、粒子群最適化 (PSO)²⁾ や Firefly algorithm (FA)³⁾ に代表される群知能のアルゴリズムでは、探索点の挙動は差分方程式に基づく力学系で記述されるが、条件分岐による自己最良点の交換などの手続きを含むことと多数の点の相互作用からなる多体系であることが挙げられる。進化計算においても、集団の世代変化を数理モデルとして記述することが可能であったとしても、その解析は同様に困難である。また、進化計算の多くでは、次世代の集団を構成する手順そのものがアドホックであり、世代交代則を数理モデルとして記述することも容易ではないと考えられる。

一方で、多体系であっても質点の数を限りなく多くすると、連続的に分布する量の濃度に近似することが可能になってくる。濃度の変化は、偏微分方程式で記述することができ、解の挙動を調べる解析的手法が確立されている。そこで我々は、進化計算の探索点の力学系が多体系であることに由来する困難さを回避するとともに世代交代の手順を数理的に記述することに代えて、探索点を偏微分方程式で記述された濃度のサン

プル点とみなす形の進化計算の手続きを検討している。これは、分布推定アルゴリズム (EDA) と同様の手続きを与えており、言い方を換えれば、通常の EDA では探索点から次世代の確率分布が更新されるのに対して、解析が十分に行われている力学系に従って確率分布が更新されていく EDA を与えるものである。

本研究では、生体内におけるナノマシンと生体分子との相互作用による拡散や凝集を扱った偏微分方程式を利用する。このモデルに基づいて、Iwasaki が提案したエージェントの振る舞いを濃度として取り扱うモデルでは、エージェントの濃度分布がターゲットとなる生体分子の濃度分布と同様の分布に収束することが示されている⁴⁾。これまでに、このことを利用して、ターゲットの濃度分布を目的関数とする最適化問題を想定し、エージェントの濃度に対応する確率密度関数に基づく EDA の枠組みを提案した⁵⁾。また、数値計算によって、提案した EDA による解探索により得られる確率密度関数の極大値と目的関数の極大値が一致することを示した。本発表では、このアプローチにおいて、もととなる偏微分方程式のモデルに含まれるパラメータの導出された EDA 型の手続きから得られる確率分布への影響を議論し、パラメータの物理的意味によって、探索の集中と拡がりを制御できることを示す。

なお、本論文は以下のように構成される。第2章では Iwasaki が提案したモデルと証明した定理の紹介を行う。第3章ではその定理を利用した EDA の枠組みを提案する。そして第4章で構成した EDA を用いて簡単なベンチマーク問題を解き、解探索性能について議論する。

2 Target-Detection Model

分子通信とは、自然界に存在するか又は生体材料から人工的に製造された微小なデバイス (生物ナノマシンと呼ばれる) が通信を行うときの一般的かつ普遍的な方法である⁶⁾。Okaie らは分子通信において、生物ナノマシンが誘引物質と忌避物質のみを用いて、外部からの信号を受けることなく自己創発的に目的の場所 (ターゲット) を検出することを想定した巨視的なモデル方程式を提案し、適切にパラメータを選ぶことによ

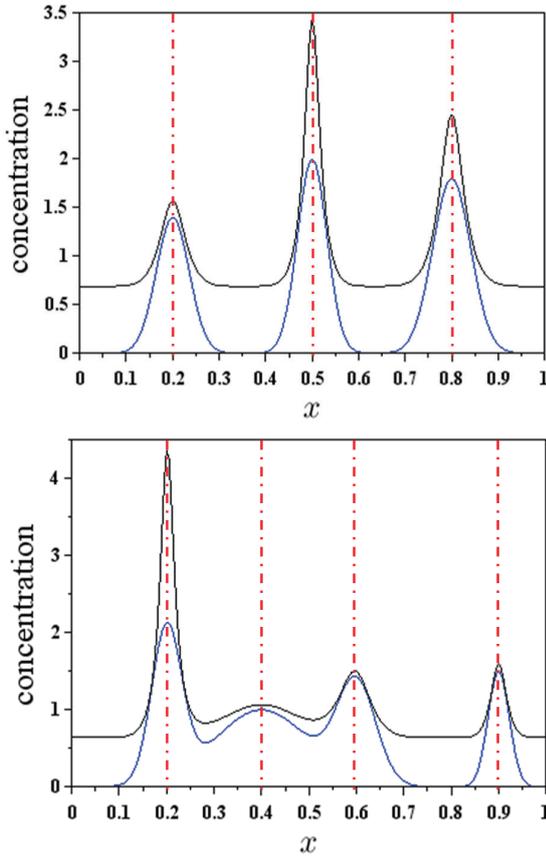


Fig. 1: Examples of targets (blue curves) and stationary distribution of nanomachines (black curves).

り、生物ナノマシンがターゲットに集中することを数値的に示した⁷⁾。

Iwasaki は、Okaie らのモデルにある誘引物質と忌避物質に対して (i) 誘引物質と忌避物質は拡散しない、(ii) 誘引物質と忌避物質は十分に短い時間で定常状態に到達する、という 2 つの仮定をして、生物ナノマシンの濃度に関する以下のモデル方程式を提案した⁴⁾。

$$\begin{cases} \frac{\partial v}{\partial \tau} = \frac{\partial^2}{\partial x^2}(av + \alpha v^2) - \mu \frac{\partial}{\partial x} \left[v \frac{\partial}{\partial x}(f(x)v) \right], \\ v(x, 0) = v_0(x), \end{cases} \quad \begin{matrix} \text{in } \Omega \times T, \\ \text{in } \Omega. \end{matrix} \quad (1)$$

ここで、 $\Omega = [0, 1]$ は探索領域であり、 $T = (0, \infty)$ は時間を表している。 $f(x) \geq 0, x \in \Omega$ はターゲットと呼ばれる物質の濃度を表し、ターゲットの濃度が濃い場所に生物ナノマシンが集中することにより、生物ナノマシンがターゲットを検出したと考える。 $v = v(x, \tau), (x, \tau) \in \Omega \times T$ は時刻 τ における Ω 上での生物ナノマシンの濃度を表している。 $v_0(x)$ は初期時刻 $\tau = 0$ における生物ナノマシンの濃度を表していて、既知の関数とする。また、 $a > 0$ はエージェントの拡散を表すパラメータであり、 $\alpha > 0$ は忌避物質の効果の強さを表すパラメータである。 $\mu > 0$ は誘引物質の効果の強さを表すパラメータであり、 μ が大きいほどターゲットへの集中が起こりやすくなるという物理的な特性がある。境界条件は反射壁条件が仮定されている。

Iwasaki により得られている定理⁴⁾ を本研究に適用できる形で述べる。

定理 $f(x) \in \mathcal{C}^3(\Omega)^1$ を非負の関数とする。さらに、パラメータ α, μ は

$$2\alpha \geq \mu \cdot \max_{x \in \Omega} f(x), \quad (2)$$

を満たすとする。このとき、反射壁条件を課した方程式 (1) の解 $v(x, \tau)$ は、以下の 3 つの性質を満たす。

I 解 $v(x, \tau)$ は

$$\int_{\Omega} v(x, \tau) dx = \int_{\Omega} v_0(x) dx, \quad \forall \tau \in T$$

を満たす。つまり、探索区間 Ω 上の生物ナノマシンの総数は、初期時刻のナノマシンの総数から変化しない。

II 解 $v(x, \tau)$ は $\tau \rightarrow \infty$ において唯一の定常分布 $\bar{v}(x)$ に収束する。

III 定常分布 $\bar{v}(x)$ とターゲット $f(x)$ の極小値と極大値の座標は一致する。

この定理の中の性質 III が表現していることを Fig.1 に図示する。ターゲット $f(x)$ (青線) と、生物ナノマシンの定常分布 $\bar{v}(x)$ (黒線) の極値の座標が一致することが分かる。

さらに Fig.2 は同じターゲットに対する、生物ナノマシンの定常分布のパラメータ μ 依存を示している。 μ が大きいほど生物ナノマシンがターゲットにより集中していることが分かる。

3 定理を応用した EDA 型の探索アルゴリズムの拡張

我々は定理の中の I の性質により、初期時刻の生物ナノマシンの濃度 $v_0(x)$ が $\int_{\Omega} v_0(x) dx = 1$ を満たすとき、その解 $v(x, \tau)$ は生物ナノマシンの存在確率の時間変化と見ることが出来ること考えた。そこで生物ナノマシンの濃度分布を、進化計算におけるエージェントの確率密度関数と見なして、2章で紹介した定理を EDA 型の探索アルゴリズムに適用することを検討している。具体的には方程式 (1) の $f(x)$ を最大化問題の目的関数と考え、方程式 (1) にしたがって確率密度分布を更新することにより、複数の局所解の近傍からサンプル点が比較的高い頻度で得られることを保証できると考えている。本発表は、このアルゴリズムにエリート保存のメカニズムを加えたアルゴリズムの性能を議論することを目的としている。また、方程式 (1) 中にある物理的なパラメータが、提案アルゴリズムの性能にどのような影響を与えるかを議論することも目的とする。

ブラックボックス型の目的関数値最大化問題に定理を応用するために、エージェントの観測した点から目的関数の情報を反映するターゲット関数を構成する、以下の手順を導入した⁵⁾。 $x_i \in \Omega, i = 1, \dots, n$, を n 個の観測点として、 $f(x_i), i = 1, \dots, n$, をその観測値とする。また、 $f_0 = \min_{i=1, \dots, n} f(x_i)$ とする。このとき、目的関数の推定値 $\hat{f}(x)$ を

$$\hat{f}(x) = \sum_{i=1}^n \pi_i N(x; x_i, \sigma), \quad x \in \Omega, \quad (3)$$

¹ $\mathcal{C}^3(\Omega)$ 関数とは区間 Ω 上で 3 階微分可能であり、3 次導関数が連続となる関数のことである。

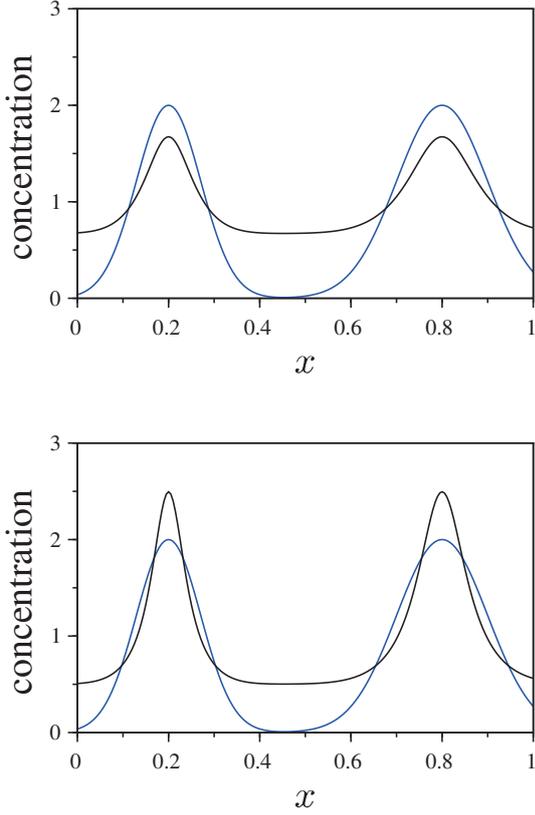


Fig. 2: Examples of targets (blue curves) and stationary distributions of nanomachine (black curves). Upper and lower figures denote the case where $\mu = 3$ and $\mu = 4$, respectively.

により与える. ここで $N(x; x_i, \varepsilon) = \exp(-\varepsilon|x - x_i|^2)$ であり $\varepsilon > 0$ は正のパラメータとする. また重み係数 π_i は

$$\pi_i = \frac{f(x_i) - f_0}{\sum_{j=1}^n (f(x_j) - f_0)},$$

により決定する. (3) のように定められた推定値 $\hat{f}(x)$ を用いることにより, 目的関数 $f(x)$ が連続でなく, 非負でなくても定理の応用が可能となる. 特に, パラメータ条件 (2) は $2\alpha \geq \mu$ が成り立てば良いことになる.

本発表で提案するアルゴリズムの手続きは以下のようによまとめられる (Fig.3 参照):

-
- Step 1.** 世代 $t \leftarrow 0$ と, 初期の確率密度関数 $u^0(x)$ を設定する.
 - Step 2.** $u^t(x)$ からエージェントの候補 \tilde{x}_i , $i = 1, \dots, n$, をサンプリングする.
 - Step 3.** $t-1$ 世代におけるエージェントと, \tilde{x}_i , $i = 1, \dots, n$, を比較して, 目的関数値が良いものから順に計 n 個選び, それを t 世代におけるエージェント x_i , $i = 1, \dots, n$, とする.
 - Step 4.** 選ばれた目的関数値 $f(x_i)$, $i = 1, \dots, n$, から目的関数の推定値 $\hat{f}(x)$ を計算する.
 - Step 5.** $f(x)$ を $\hat{f}(x)$ に置き換えた方程式 (1) にしたがって確率密度関数を更新する, $u^{t+1}(x) \leftarrow u^t(x)$.
 - Step 6.** 世代を更新する, $t \leftarrow t+1$.

Step 7. 打ち切り条件を満たしていなければ, Step 2 に戻る.

Step 8. 最終的な確率密度関数 $u^t(x)$ を出力する.

Step 5 における確率密度関数の更新は, 偏微分方程式の差分法による数値計算を用いる. 数値計算において, 時間の分割幅を $\Delta\tau$, 空間の分割幅を Δx とする. この際, 初期確率密度分布を $u^t(x)$ と設定して, $f(x)$ を $\hat{f}(x)$ に置き換えた方程式 (1) にしたがって, 数値計算の打ち切り時間 $\tilde{\tau} > 0$ までの間, 数値計算を行う.

本アルゴリズムにおいて定理によって理論的に保証されている部分は, 数値計算の打ち切り時間 $\tilde{\tau}$ が十分に大きければ, $u^{t+1}(x)$ と $\hat{f}(x)$ の極小値と極大値の座標が一致するというのである.

従来のアルゴリズム⁵⁾ との違いは, Step 3,4 にあるようにエリートのみを用いて目的関数の推定を行うことである. この手順を導入することで, 世代が進むにつれて目的関数値が良い座標で極大値を持つ推定値 $\hat{f}(x)$ が得られるので, 局所解近傍でより集中的なサンプリングが行われると期待できる.

4 数値例

本章では, 数値計算を行った結果を示す. 特に, 提案アルゴリズムのパラメータ依存を調べるために, 本発表では誘引物質に関するパラメータ μ を変えた結果を示す. Fig. 2 にも示したように, μ が大きいほどエージェントが集中しやすいことに注意する.

各世代でのサンプル点の数, 及びエリートとして残るサンプル点の数は $n = 10$ とする. 方程式 (1) のパラメータは $a = 0.01$, $\alpha = 1$ とする. 目的関数の推定に用いるパラメータは $\varepsilon = 50$ とする. 方程式 (1) の差分による数値計算は $\Delta\tau = 1 \times 10^{-6}$, $\Delta x = 1/100$, $\tilde{\tau} = 1 \times 10^{-4}$ により行った. また初期の確率密度分布 $u^0(x)$ は探索領域 Ω 上の一様分布とする.

目的関数として以下の4つの関数を用いた. 微分可能で2つの峰を持つ関数

$$f_1(x) = e^{-500(x-0.25)^2} + 2e^{-500(x-0.75)^2}$$

と, ほぼ平坦な中に小さな峰が存在する関数

$$f_2(x) = 1 - e^{-10x} + \frac{1}{10}e^{-1000(x-0.8)^2}$$

と, 不連続な関数

$$f_3(x) = \begin{cases} 0, & (x < 0.5) \\ e^{-10(x-0.5)}, & (0.5 \leq x) \end{cases}$$

と, 階段状で左端に大域最適解が存在する関数

$$f_4(x) = -\frac{1}{10}[10x] - \frac{1}{100}x + 2$$

を用いる.

Fig. 4 に目的関数が $f_1(x)$ の場合の, 確率密度関数の時間発展の様子を示す. 各図において, 赤線は真の目的関数 $f(x)$, 緑線は各世代における目的関数の推定値 $\hat{f}(x)$, 黒線は各世代におけるエージェントの確率密度関数 $u^t(x)$, 青点はその世代までにおけるエリートのエージェントを表している. どの μ においても, 世代

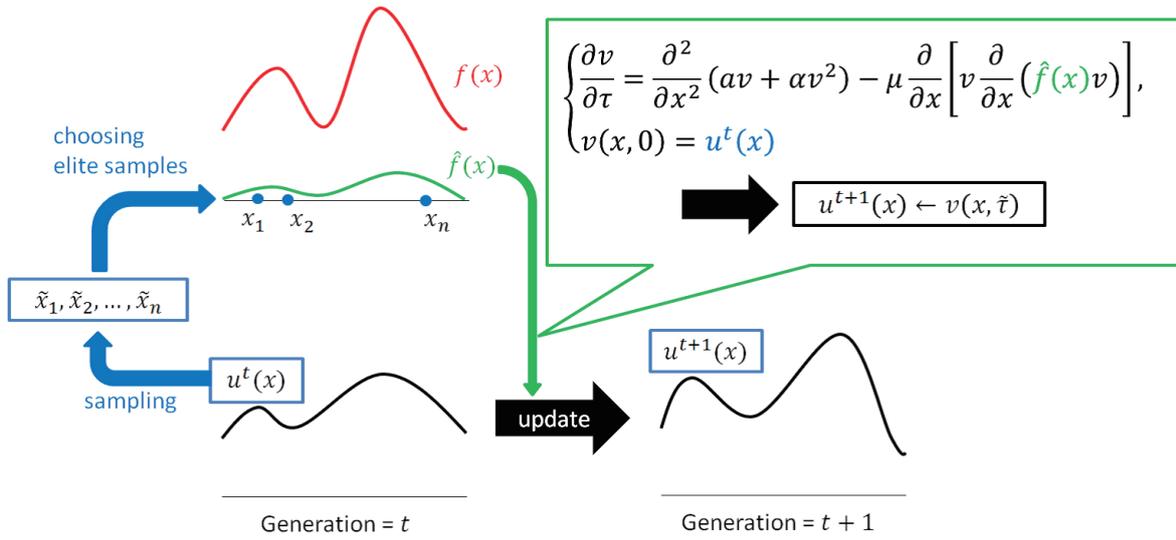


Fig. 3: Procedures of our proposed algorithm

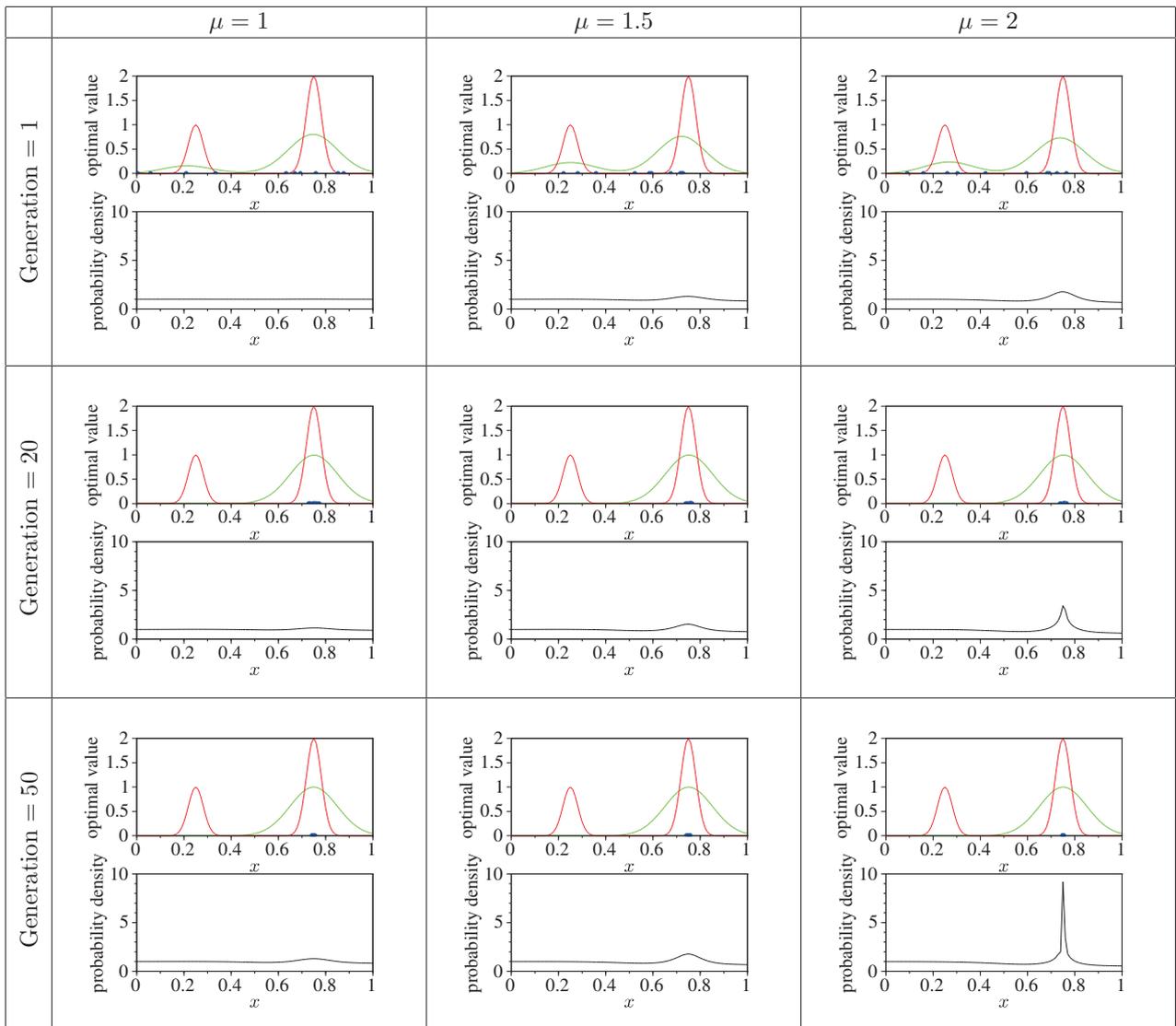


Fig. 4: Numerical results for $f_1(x)$

が進むにつれて大域最適解の近傍で確率密度が大きい確率密度関数に変化している。さらに μ が大きいほど、確率密度関数がより集中したサンプリングが行える関

数に変化していることが分かる。これは μ の方程式 (1) の中での物理的な意味から考えると、合理的な結果となっている。

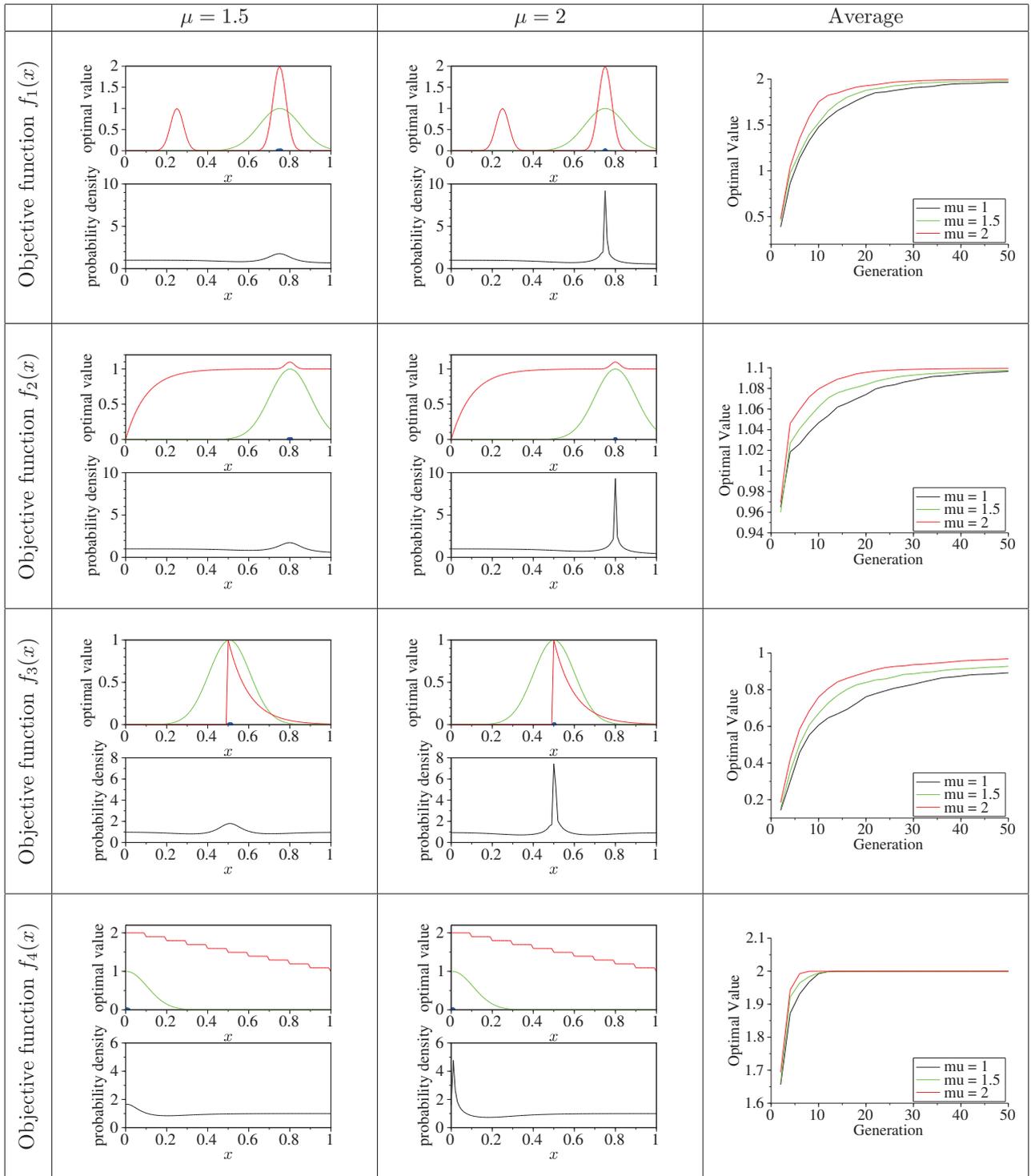


Fig. 5: Numerical results for each objective function

Fig. 5 に各目的関数の、50 世代における確率密度関数の様子と、各世代における目的関数値の平均値を示す。平均値に関しては、各 μ ごとに 10 回の試行を行い、各世代ごとにその平均を取った。どの目的関数においても、 μ が大きい方が大域最適解の近傍でより集中したサンプルが行える確率密度関数に変化している。 $f_3(x)$ や $f_4(x)$ のように、不連続な関数や探索区間の端に最適解があるような目的関数においても、大域最適解を探索できている様子が分かる。目的関数値の平均値を見ると、どの目的関数においても、世代が進むに

つれての平均値の向上率は μ が大きいほど良い。結果として、本発表で用いた目的関数においては、 μ が大きいほど大域最適解を探索する能力が高いことが分かる。

5 おわりに

我々は、生体内の分子通信の分野で提案された生体分子と相互作用するエージェントの振る舞いを濃度として取り扱う偏微分方程式モデルに基づき、その解析結果を利用した EDA 型の探索アルゴリズムを与えている。本論文では、進化計算におけるエリートに対応するエージェントを選ぶことにより、探索性能の改善

を試みた。さらに、モデルに含まれる物理的なパラメータが、提案アルゴリズムの振る舞いに与える影響を調査するための数値計算を行った。数値実験によって、提案法のアルゴリズムに従って更新される確率密度関数は、大域最適解の近傍で比較的高い確率密度を持つことが示された。また、パラメータを調整することにより、最適解の近傍での集中的なサンプリングとなるように調整することができた。他の目的関数の場合や、別のパラメータを調整した場合についても、同様に調査を進めており、機会を改めて結果を報告する。

謝辞

本研究の一部は、文部科学省博士課程教育リーディングプログラムと、JSPS 科研費 JP15K00338, JPKT0105 の助成を受けた。ここに謝意を表す。

参考文献

- 1) 田邊遼司, 串田淳一, 島中利治: 関数最適化における進化計算, 計測と制御, **Vol. 54** No. 8, 567/572 (2015)
- 2) Kennedy, J. and Eberhart, R.: Particle swarm optimization, Proceedings of the IEEE International Conference on Neural Networks, **Vol. 4**, 1942/1948 (1995)
- 3) Yang, X.-S.: Firefly algorithms for multimodal optimization, Stochastic algorithms: foundations and applications, **Vol. 5792**, 169/178 (2009)
- 4) Iwasaki, S.: Convergence of Solutions to Simplified Self-Organizing Target-Detection Model, submitted to *Scientiae Mathematicae Japonicae*.
- 5) 岩崎悟, 島中利治: 誘引・忌避モデルの解析とモデルに基づく分布推定アルゴリズムに関する考察, 第 11 回進化計算学会研究会資料集, 115/120 (2016)
- 6) 中野賢, Michael Moore, 榎本章宏, 須田達也: 分子通信技術の研究動向 バイオ ICT としての分子通信, 情報通信研究機構季報, **Vol. 54**, No. 4, 73/88 (2008)
- 7) Okaie, Y., Nakano, T., Hara, T., Hosoda, K., Hiraoka, Y., and Nishio, S.: Modeling and performance evaluation of mobile bionanosensor networks for target tracking, Proceedings of the IEEE International Conference on Communications, 3969/3974 (2014)

多様なエージェントによる集団強化学習の

自動帆走への応用

○橋完太, 深沢竜大
工学院大学情報学部

Ensemble Reinforcement Learning by Diverse Agents and its Application to Autonomous Sailing

*Kanta Tachibana and Ryuta Fukazawa

Faculty of Informatics, Kogakuin University

Abstract

This paper discusses effect of ensemble algorithm for reinforcement learning of garbage collecting sailing. Robotic sailing is a promising solution for semi-permanent collecting garbage float on oceans. Reinforcement learning makes agents to learn intelligent actions in complex environment like on oceans. Effects of ensemble learning have not been studied yet well in combination with reinforcement learning. Through numerical simulations, we found better decision making performance with ensemble learning.

1. はじめに

Q学習やActor-Critic学習などの強化学習は、教師信号無しに行動を学習する手法として提案された[1, 2, 3]。強化学習によって、エージェントは、得られる報酬に応じてさまざまな状態に合わせて行動の価値を更新する。強化学習では、事前に制御ルールを与える必要がなく未知の環境でも学習を進められるため、多くの制御問題に応用されている[4]。しかし、強化学習は「次元の呪い」に苦しめられる。つまり、状態空間の次元が大きくなるにしたがって、強化学習の結果、局所解に収束してしまい、エージェントは望ましくない行動を繰り返すことになる[5]。

倒立振り子[6]や立ち上がり動作[7]、ルート探索[8]などのロボット制御の問題は多くの場合、連続な状態空間で強化学習を行わなければならない、状態空間をどのように離散

化するかがカギを握る。状態変数をファジィ化したファジィQ学習[9, 10]が提案され、観測誤差などを吸収するファジィな状態空間の離散化が実現されている。他の離散化手法として、状態空間の自律的な再構成手法[11]と適応スイッチングコントローラ[12]が提案されている。将来の地図(環境)が確率的にしか予測できない帆船の舵制御[13, 14]は部分観測マルコフ過程(POMDPs)であり、他の制御問題よりも難しい。

集合学習[15]は、教師付学習に対する強力な学習スキームとして提案された。集合学習では弱学習器の集合が用いられる。弱学習器は、一部の学習データのみで学習したり、一部の観測変数のみを用いて学習する。弱学習器からの出力が多数決などで集計され、最終的な出力が決定される。集合学習

による最終的な出力は、全てのデータ、全ての観測変数を用いて学習する強学習器に比べて、優れた意思決定を行う。Sugiyamaら[16]は集合学習の有効性をコンピュータ将棋で示した。弱学習器の重み（票数）を適応的に調整する Adaboost[17]も有効な集合学習手法である。

Wieringら[18]は、強化学習問題に集合学習を応用した。[18]では、それぞれの弱学習器が Q 学習、Actor-Critic 学習、サルサ学習など異なる学習様式を採用し、それらの集合学習が単一エージェントによる学習よりも高い性能を示した。つまり、各エージェントが異なる学習モデルを持つ集合強化学習を行った。

著者は、使用する観測変数が互いに異なるエージェントによる集合学習を自動ヨットへ適用した[19]。自動ヨットでは、風を推進源とするため、急激な風向の変化に迅速に対応しなければならない。先行研究[18]と異なり、エージェントは全て Q 学習を用い、一部の観測変数のみを用いる。本研究では、[19]で行った数値シミュレーションの結果について、障害物回避などの新たなタスクへの展望を示す。

第 2 節では、自動ヨットのシミュレーション設定と観測変数を説明し、第 3 節で多様なエージェントによる集団強化学習とその自動ヨットシミュレーションへの応用を説明する。第 4 節で数値シミュレーションの結果を示し、第 5 節で集合学習におけるエージェントの貢献度および障害物回避などのタスクについて考察し、第 6 節でまとめと今後の展望を述べる。

2. 帆船シミュレーションと観測変数

2.1 帆船シミュレーション

2 次元平面内を移動する帆船を想定し、風速が一定で風向がランダムに変化するシミュレーションを実装する。シミュレーションでは、帆船が風を受けて進み水の抵抗を受けて減速することを想定する。帆船のすすむ方向をロボットが決めて、与えられた目的地点に早く到達できることを目標とする。風向は定期的に $\pm 5^\circ$ の範囲の一様乱数に従って変化することとする。平面内に目標地点を定め、帆船の操縦者は、自身から見た目標物の方角を正確に認識できるものとする。また、帆船の操縦者は、自身の帆船が向いている方角と、風向も、誤差無しで認識できるものとする。帆船が目標位置に到達すると、1 エピソードが終了となり、次のエピソード用に新たな目標位置が設定される。

帆船が受ける力は、帆にあたる風の強さの二乗と、帆が風を受ける面積に比例するものとし、そのうちの推進成分が推進力を生み出すと想定する。左右成分は、水から受ける抗力により打ち消されるものとする。また、推進方向には、艇速の二乗に比例した抵抗がかかるものとする。操縦者は、舵だけを操縦し、帆は、推進力が最大となるように制御されるものとする。

帆船の操縦者は、従来の強化学習、または、提案する集合学習によって行動を学習する。操縦者が、舵を切る行動を選択した場合には船の方角が 3° 変化するものとする。

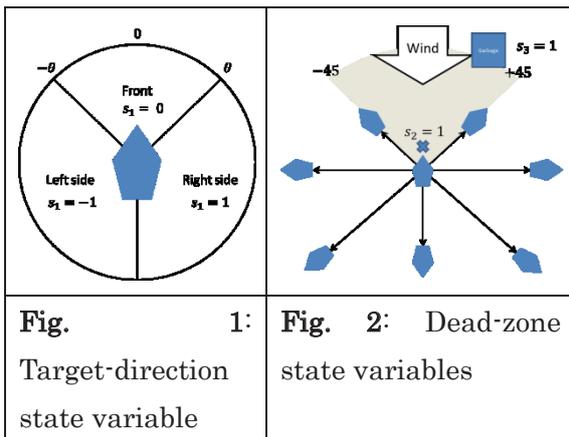
2.2 観測変数

帆船の操縦者は、3 種類の観測変数を利用できるものとする。1) 船の進行方向に対する目標位置の方角 s_1 ; 2) 風向に対する船の進行方向 s_2 ; 3) 風向に対する目標位置の方

角 s_3 、の 3 つの観測変数を利用できるものとする。それらは、次のように離散化される。

- 1) 目標位置が右舷側 ($s_1 = 1$), 左舷側 ($s_1 = -1$), 前方 ($s_1 = 0$)。
- 2) 船がデッドゾーン (風向に対して 45° より鋭角になる方向) を向いている ($s_2 = 1$), デッドゾーンを向いていない ($s_2 = 0$)。
- 3) 目標位置がデッドゾーン (風向に対して 45° より鋭角になる方向) に入っている ($s_3 = 1$), デッドゾーンに入っていない ($s_3 = 0$)。

観測変数 s_1 の離散化のために、離散化パラメータ θ を導入する (図 1)。また、図 2 に残りの 2 つの観測変数を示す。



3. 自動ヨットの集合強化学習

集合学習では、多数のエージェントの下した意志決定を集積する。先行研究[18]では、意志決定の集積方法についても検討された。ボルツマン和やボルツマン積、加重平均、加重幾何平均などを比較し、ボルツマン積による集積が優れることを示した。本研究でも一人との下した意志決定の集積をボルツマン積で行うこととする。

集合学習に参加するエージェントには、多様性を持たせる。アルファ・エージェントと呼

ぶエージェントは、3 つの観測変数全てを用いて、事前に強化学習をある程度進めておく。その他のエージェントは、1 つまたは 2 つの観測変数を用い、強化学習のパラメータである学習率と割引率や、観測変数 s_1 を用いるエージェントでは離散化パラメータ θ が異なる。

図 3 に帆船シミュレーションおよび提案手法の流れ図を示す。

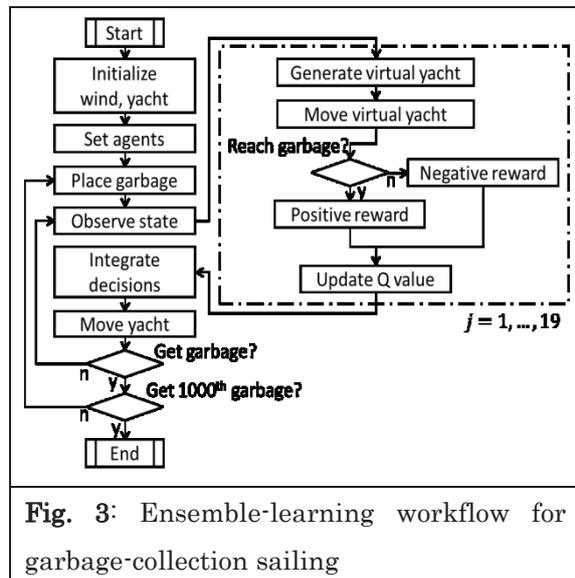


Fig. 3: Ensemble-learning workflow for garbage-collection sailing

まず、帆船の位置・方角と風を初期化し、エージェントの Q 表を初期化する。次に目標地点を設定し、各エージェントが観測を行い離散化する。提案手法では、各エージェントが仮想ヨットを実際のヨットと同じ位置・方角・速度で生成し、各エージェントの Q 表からソフトマックス法で選択した行動に従って、仮想ヨットを動かす。目標位置に到達していれば正の報酬を、そうでなければ負の報酬を受け取り、 Q 値を更新する。各エージェントが選択した行動は、ボルツマン積によって集積されて、実際のヨットの行動を決定する。

4. シミュレーション結果

帆船シミュレーションにおける集合学習の効

果を明らかにするため、2段階でシミュレーションを行った。1段階目は従来の強化学習による、帆船操縦の強化学習。2段階目は、従来の強化学習と提案する集合強化学習との比較である。

1段階目のシミュレーションでは、3つの観測変数全てを用いるエージェントの強化学習を行う。状態空間を12の状態に離散化する。このエージェントをアルファ・エージェントと呼び、Q学習における学習率 $\alpha = 0.001$ 、割引率 $\gamma = 0.999$ 、離散化パラメータ $\theta = 45^\circ$ 、ソフトマックスの温度パラメータ $T=0.7$ とする。

2段階目のシミュレーションでは、2艇の帆船で競争を行い、集合学習アルゴリズムの効果を明確にする。ただし、他艇や障害物との衝突を考慮すると衝突回避のための観測変数が増えるため、2艇の衝突や干渉は考慮しないこととする。衝突の回避については、第5節で議論する。1艇はアルファ・エージェントにより制御され、もう片方の1艇は、アルファ・エージェントを含む19エージェントの集合学習により制御される。図4に、シミュレーションのスクリーンショットを示す。また、表1に1000エピソードの勝敗表を示す。

Table 1: Standings of the competitions

	Wins	Losses	Draws
Ensemble	385	200	415
Single Agent	200	385	415



Fig. 4: Competitions between a single-agent and an ensemble of agents.

5. 考察

表1に示した1000エピソードの勝敗数は集合学習の勝数が有意に多かった。集合学習に参加した19エージェントの貢献度を調べる。各エージェントが最終的に得たQ表は、状態行動対の価値を表している。つまり、各状態行動対のQ値は将来得られる報酬が逆伝搬して上下するため、良い選択をしたエージェントの状態行動対はQ値が高くなる。1000エピソード終了後に、一定期間、ランダムにシフトする風の中で目標位置へ到達するタスクを行い、その間の各エージェントが認識・選択した状態行動対のQ値を合計し、エージェントの貢献度とする。アルファ・エージェントは、全ての観測変数を用い、集合学習開始前にある程度学習を行っていたため、当然、貢献度が大きくなる。他の18エージェントでは、学習パラメータと離散化パラメータがアルファ・エージェントと同一に設定されたエージェントが、アルファ・エージェントと異なるパラメータに設定されたエージェントよりもQ値が高かった。それら6エージェントのうち、用いた観測変数に注目すると、① s_3 の

み、② s_2 のみ、③ s_1 と s_2 、④ s_1 のみ、⑤ s_2 と s_3 、⑥ s_1 と s_3 の順位となった。観測変数が1つだけのエージェントは、状態数が2または3となるため、ひとつの状態がより頻繁に認識され、Q値の変動が大きくなる。一つの観測変数を用いる場合には、風との相対方向である s_3 や s_2 を用いたエージェントの貢献度が大きかった。

今後、シミュレーション上にて、固定障害物や他艇との衝突を回避する機能を持たせる予定である。その際、方角ごとの障害物の有無やその距離、接近中の障害物が移動するか否かを状態変数として持たせる必要がある。また、舵の操作（右に舵を切る／左に舵を切る／舵を切らない）の他に、自艇を加速する／加速しない、の行動も考慮する必要がある。その際、状態空間および行動空間の次元が増大するため、本研究で提案したような一部の変数のみを観測するエージェント、また、一部の行動変数のみに投票するエージェントによる集合学習が有効と期待できる。

6. まとめ

多変量で連続な状態空間を持ち、環境が時々刻々変動する強化学習問題に対し、一部の変数のみを観測するエージェントによる集合強化学習アルゴリズムを提案し、その有効性を帆走の数値シミュレーションにおいて示した。今後、衝突回避などの複数のタスクを同時に達成するため、より多くの変量を想定し、提案手法の有効性を明らかにしたい。

参考文献

[1] C. Watkins, "Learning from

Delayed Rewards," PhD thesis, Cambridge University, Cambridge, 1989.

[2] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," MIT Press, 1998.

[3] A. G. Barto, R. S. Sutton, and W. C. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," IEEE Trans. Syst., Man., Cybern., Vol. SMC-13, Issue 5, 834-846, 1983.

[4] M. Asada, S. Noda, S. Tawaratsumida, and K. Hosoda, "Purposeful behavior acquisition for a real robot by vision-based reinforcement learning," Machine Learning, 23, 279-303, 1996.

[5] A. Notsu, H. Ichihashi, and K. Honda, "State and action space segmentation algorithm in Q-learning," Int'l Joint Conf. on Neural Networks, 2384-2389, 2008.

[6] Y. Kashimura, A. Ueno, and S. Tatsumi, "A Continuous Action Space Representation by Particle Filter for Reinforcement Learning," 22nd Annual Conf. of the Japanese Society for Artificial Intelligence, 2008 (in Japanese).

[7] J. Morimoto and K. Doya, "Reinforcement learning of dynamic motor sequence: Learning to stand up," In Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, Vol. 3, 1721-1726, 1998.

[8] S. Ishihara and H. Igarashi,

- “Applying the policy gradient method to behavior learning in multiagent systems: The pursuit problem,” *Systems and Computers in Japan*, Vol. 37, Issue 10, 101-109, 2006.
- [9] P. Glorennec, I. d. R. F. Dept. d’Inf., and L. Jouffe, “Fuzzy Q-learning,” In *Proc. Sixth IEEE Int’l Conf. on Fuzzy Systems (Volume: 2)*, 1997.
- [10] Y. Hosoya and M. Umamo, “Dynamic fuzzy Q-learning with facility of tuning and removing fuzzy rules,” *IEEE World Congress on Computational Intelligence*, 2012. [11] T. Yairi, K. Hori, and S. Nakasuka, “Autonomous reconstruction of state space for learning of robot behavior,” *Proc. of Int’l Conf. on Intelligent Robots and Systems*, 891-896, 2000.
- [12] M. Nagayoshi, H. Murao, and H. Tamaki, “A reinforcement learning with switching controllers for a continuous action space,” *Artificial Life and Robotics*, 15, 97-100, 2010.
- [13] P. J. Sterne, “Reinforcement Sailing,” *Master of Science, Artificial Intelligence, School of Informatics, University of Edinburgh*, 2004.
- [14] H. Manabe and K. Tachibana, “Consideration of state representation for semi-autonomous reinforcement learning of sailing within a navigable area,” *Robotic Sailing*, 89-102, 2015.
- [15] N. Ueda and R. Nakano, “Generalization error of ensemble estimators,” *IEEE Conf. on Neural Networks*, 90-95, 1996.
- [16] T. Sugiyama, T. Obata, K. Hoki, and T. Ito, “Optimistic selection rule better than majority voting system,” *Computers and Games, LNCS 6515*, 166-175, 2011.
- [17] Y. Freund and E. R. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, Vol. 15, Issue 1, 119-139, 1995.
- [18] M. A. Wiering and H. V. Hasselt, “Ensemble algorithm in reinforcement learning,” *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 930-936, 2008.
- [19] K. Tachibana and R. Fukazawa, “Effect of an Ensemble Algorithm in Reinforcement Learning for Garbage-Collection Sailing”, *Robotic Sailing 2016*, pp.79-90

不確実性を考慮した遺伝的アルゴリズムによるバスダイヤの生成

○沼田 雅史 橋 完太 (工学院大学)

Bus Timetable Generation by Genetic Algorithm Considering Uncertain Costs

* M. Numata and K. Tachibana (University of Kogakuin)

Abstract— Bus timetable has been made by hand. Recently, some approaches have been developed to generate bus timetable automatically. Among them, genetic algorithm is often applied to this problem. However, most of them set costs as fixed values. In practice, gap occurs between actual time and estimated time because of time-varying and uncertain traffic congestion. Therefore, costs change probabilistically. Although there are studies considering stochastic time, it is not adequate in Japan as they assume earlier departure than schedule. In this study, we propose a new genetic algorithm considering uncertain costs and apply this algorithm to bus timetable generation problem.

Key Words: Bus Timetable, Genetic Algorithm

1. はじめに

バスダイヤ生成は、車両や運転手のスケジューリング、運行コストの削減、輸送客数の増加など、様々な要素を考慮しなくてはならず、複雑な問題である。バスダイヤの自動生成に関する研究はこれまで数々行われてきたが、その中でも遺伝的アルゴリズム(以下 GA とする)は多目的最適化問題に適していることから、使用されることが多い¹⁾²⁾。

GA を用いる研究の多くは、バス停間の距離や所要時間を一定の値として扱うことで不確定なバスダイヤを生成しているが、実際には道路の混雑状況等により所要時間は変化するため、それに合わせた時間設定が必要である。そこで Wu ら³⁾は、所要時間が確率的に変化するという条件下において、予定時刻にゆとりの時間を設けることで、乗客が被る待ち時間を減らすバスダイヤの生成を行った。しかし、Wu らはバスの早発を前提としているが、日本においては早発は法律で禁止されており、バス停に予定より早く到着した場合、予定時刻まで待機しなければならない。この制約に従うと、ゆとりの時間を設けることは遅延を減らすことには有効でも、待機時間をかえって増加させることに繋がり、最適手法ではない。

また、期待値のみで個体を評価すると、例えば期待値は優秀でも実は分散が大きく、遅延がほとんど起きない時と、長時間の遅延が発生する場合との差が激しい個体も高く評価されてしまう。そのため、期待値だけでなく分布を考える必要がある。そこで、個体集団から分布として評価する手法を提案する。

本論文では、早発をしないという前提のもとでのバスダイヤを考える。また、より現実的な所要時間の変化を再現するために、道路の混雑度から車両速度を予測するモデル化を行い、そのモデルによって所要時間を確率的に与えることでシミュレーションを行う。そして不確実性を考慮した GA を用いて、遅延の少なくなるバスダイヤの生成を行う。

2 章でバス運行シミュレーションを行う上での事前準備を行う。3 章で路線候補を選定するための 1 回目の GA について説明する。4 章でバスダイヤを

生成する 2 回目の GA について説明する。2 回目の GA では、期待値のみで個体を評価する従来の GA、もしくは目的変数の確率分布で個体を評価する提案手法を用いる。5 章で処理結果を示した後に比較を行い、6 章でまとめを記す

2. バス運行シミュレーション

2.1. 問題設定

処理を行うにあたり、各種設定を行う。まず、バスを運行するネットワークとして、Fig. 1 に示す 16 × 16 のネットワークを仮定する。各点がバス停を表しており、その左上の数字がバス停番号である。バス停 0 を始点、バス停 15 を終点とする一方向のバス路線を考える。赤色の数字はバス停間の距離(km)、青色の数字は予想交通量(台/h)、緑色の数字は交通容量(台/h)であり、各道路においてこれらの値は既知として割り当てる。さらに各道路に制限速度を設定し、破線が 30km/h、実線が 40km/h、二重線が 50km/h とする。

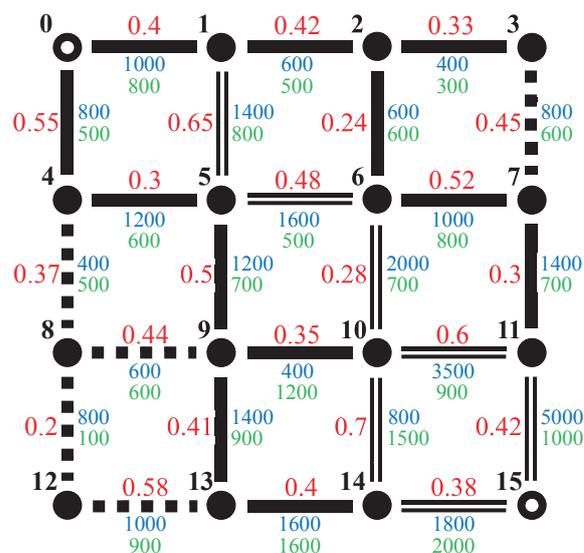


Fig. 1 : Postulated network.

Table 1 : Number of people want to go from yellow depot to cyan depot.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	-	0	0	2	0	3	1	2	1	11	1	3	4	2	2	21	53
1	-	-	2	0	1	1	0	4	0	1	0	1	1	2	0	11	24
2	-	-	-	0	0	0	0	1	0	2	1	0	0	0	0	2	6
3	-	-	-	-	1	1	0	3	0	1	2	3	1	2	0	9	23
4	-	1	0	3	-	2	1	0	3	2	0	0	2	1	2	5	22
5	-	1	0	0	0	-	0	2	0	1	0	0	0	0	0	7	11
6	-	0	0	0	2	2	-	1	0	5	0	0	0	3	1	6	20
7	-	-	-	-	0	1	2	-	2	3	0	2	2	1	0	10	23
8	-	2	0	0	-	2	1	0	-	3	0	0	0	2	0	5	15
9	-	0	0	1	0	0	1	0	0	-	0	0	1	0	0	1	4
10	-	0	0	0	1	2	0	0	1	0	-	0	1	1	0	3	9
11	-	-	-	-	0	0	1	-	0	4	0	-	0	0	1	2	8
12	-	0	0	0	-	1	2	1	-	3	2	0	-	1	0	6	16
13	-	1	0	1	-	0	2	0	-	1	0	3	-	-	0	7	15
14	-	0	1	0	-	3	0	1	-	2	0	0	-	-	2	9	
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	
	0	5	3	7	5	18	11	15	7	39	6	12	12	15	6	97	258

次に、Table 1 に示す各バス停間の需要表を仮定する。1つのバス停には1回しか停車できないため、一方向という設定上、あるバス停から次に進むことのできないバス停が存在するが、そこには-を記入している。この表は黄色のバス停で乗って水色のバス停で降りたい人数を表していて、例えば0から15へ行きたい人は21人となる。また、黄色の各バス停の横1行を乗車需要、水色の各バス停の縦1列を降車需要とする。

2.2. 混雑度を加味した速度モデル

バスダイヤの評価をするにあたり、バスの運行シミュレーションを行う必要がある。そこで、より現実的な時間経過を再現するために、平成22年度道路交通 CENSUS のデータを用いて速度のモデル化を行う。

道路は車線数、車道幅員、信号交差点などの交通特性を考慮し、1時間あたりの走行可能車両数(交通容量)が設定される。また、交通容量Cに対する実際の交通量Vの割合より、混雑度xが算出される。

$$x = \frac{V}{C} \quad (1)$$

道路交通 CENSUS のデータでは、全国の道路の午前7時~午後7時の12時間の調査で計測された混雑度、平均旅行速度などが提供されている。これらのデータから、制限速度ごとに混雑度と速度の関係を調べる。ここでは制限速度40km/hの33670件のデータでの分析を例に挙げる。

まず、全ての混雑度を小数点以下第2位で四捨五入し、平均旅行速度のデータを混雑度ごとに分ける。各混雑度において平均旅行速度が最大となる値をプロットすると、Fig. 2に示すように指数近似曲線(式2)を得ることができる。なお、サンプル数が200以上のデータのみを対象とした。

$$smax_{40}(x) = 71.314e^{-0.111x} \quad (2)$$

次に、混雑度ごとの平均旅行速度のデータをヒス

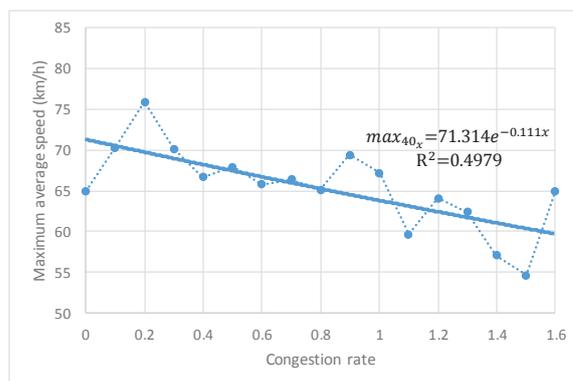


Fig. 2 : Exponential curve fitting from maximum average speed and congestion rate.

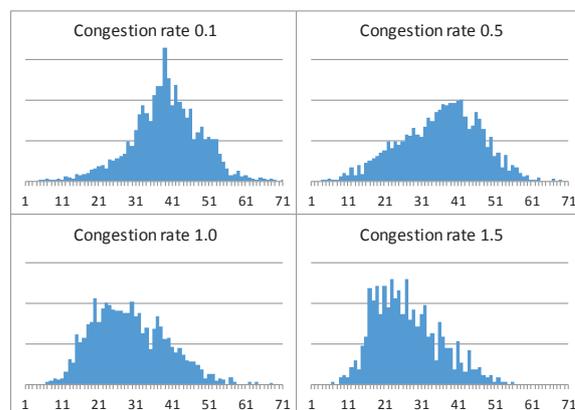


Fig. 3 : Histogram of average speed. (congestion degree:0.1, 0.5, 1.0, 1.5)

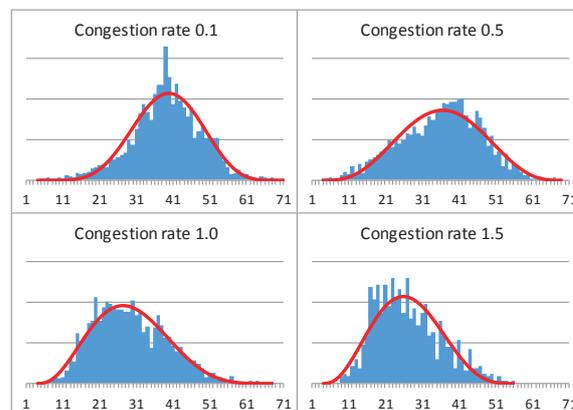


Fig. 4 : Histogram and beta distribution fitted that.

トグラム化する。Fig. 3に例として混雑度0.1, 0.5, 1.0, 1.5のデータを示す。これらのヒストグラムは、Fig. 4に示すようなベータ分布と近似させることができる。このベータ分布は、最小値を3、最大値をその混雑度の最大平均旅行速度とし、パラメータの α と β は、コルモゴロフ・スミルノフ検定の検定統計量が最小となる α と β を用いている。コルモゴロフ・スミルノフ検定の結果をTable 2に示す。ここで求めた α と β を混雑度ごとにプロットすると、Fig.

Table 2 : Calculation of α and β that beta distribution approaches to histogram by KS-test.

x	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	1.6
α	6.81	7.1	7.25	5.65	4.13	4.21	3.7	3.51	3.16	3.56	3.64	3.34	3.79	3.7	3.57	3.57	4.58
β	6	6.33	7.23	4.93	3.66	4.19	3.7	3.91	4	5.24	5.61	4.42	6.07	5.8	4.88	4.53	8.11
max	64.95	70.3	75.95	70.15	66.7	67.9	65.8	66.35	65.15	69.4	67.2	59.65	64.1	62.4	57.05	55	65
D/n	0.722*	1.265*	1.273*	1.612**	1.639	1.301*	1.451**	1.13*	0.753*	0.767*	0.823*	0.966*	1.168**	1.03*	1.437**	1.025*	0.535**

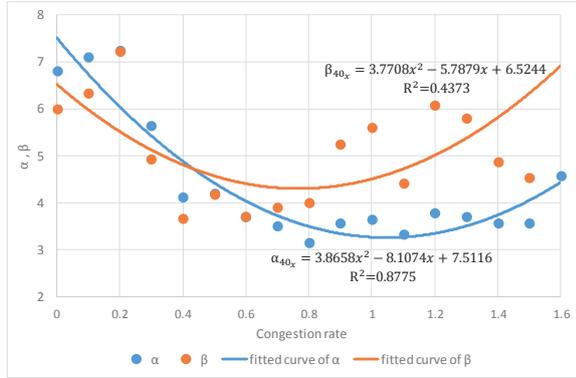


Fig. 5 : Fitted curve of α and β .

5に示すように近似曲線を得ることができる。

$$\alpha_{40}(x) = 3.8658x^2 - 8.1074x + 7.5116 \quad (3)$$

$$\beta_{40}(x) = 3.7708x^2 - 5.7879x + 6.5244 \quad (4)$$

混雑度 x が求まると、 $smax_{40}(x)$ より最大速度、 $\alpha_{40}(x)$ と $\beta_{40}(x)$ よりベータ分布のパラメータ α と β を得ることができる。このモデルを用いて混雑度から速度の分布を算出した結果を Fig. 6 に示す。橙色の線が混雑度 x の時の最大値 $smax_{40}(x)$ である。青色の線が、最小値 3、最大値 $smax_{40}(x)$ 、パラメータ $\alpha_{40}(x)$ と $\beta_{40}(x)$ からなるベータ分布の期待値であり、黒色の点が道路交通 CENSUS のデータから得た実際の平均旅行速度である。混雑度 1.6 までの実際の旅行平均速度とほぼ一致し、それ以降の混雑度においても旅行速度が予測される。

同様に制限速度 30km/h と 50km/h についてもモデル化を行った。その結果、

制限速度 30km/h の場合

$$smax_{30}(x) = 76.64e^{-0.056x} \quad (5)$$

$$\alpha_{30}(x) = 3.9018x^2 - 7.6287x + 7.4983 \quad (6)$$

$$\beta_{30}(x) = 3.2201x^2 - 3.8458x + 4.9432 \quad (7)$$

制限速度 50km/h の場合

$$smax_{50}(x) = 78.097e^{-0.093x} \quad (8)$$

$$\alpha_{50}(x) = 2.0773x^2 - 5.2659x + 7.0616 \quad (9)$$

$$\beta_{50}(x) = 0.7403x^2 - 0.7573x + 4.3992 \quad (10)$$

となる。

本研究のシミュレーションでは、走行中の道路の

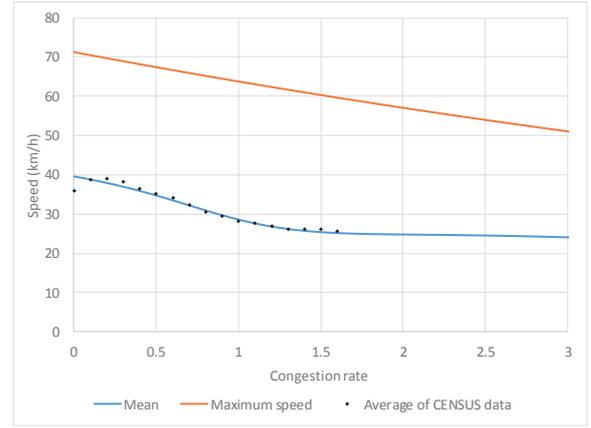


Fig. 6 : Compare this model and CENSUS data.

制限速度のベータ分布に従う実現値を生成し、それをバスの速度とし、次のバス停までの所要時間を算出する。

3. 路線候補の選定

3.1. GA による処理

本研究では GA による処理を 2 回行う。1 回目の GA では、始点 0 から終点 15 までの経路探索問題を解き、より多くの人数を輸送できる路線候補を求め、個体はバス停の停車順序を表す。個体表現には稲垣⁴⁾の提案した手法を用いる。

路線候補が Table 1 で与えた需要をどれだけ満たせるかを評価する。例えば、バス停 0, 1, 2, 3, 7, 11, 15 の順に停車する路線候補が満たせるバス停 7 における乗車需要は、バス停 11, 15 での降車数を合計した 12、降車需要はバス停 1, 2, 3 での乗車数を合計した 8 となる。また、乗車需要の高いバス停に早く停車し、降車需要の高いバス停に遅く停車するほど、その路線により輸送できる人数が多くなる。そこで評価値として、早く停車するバス停ほど乗車需要に重みを付けた値 D_1 、遅く停車するバス停ほど降車需要に重みを付けた値 D_2 を算出し、それらを加算した値 D を用いる。

$$D = D_1 + D_2 = \sum_{i=0}^{n-1} d_{on}^i(n-i) + \sum_{i=1}^n d_{off}^i(i-1) \quad (11)$$

n : その路線候補が停車するバス停数

d_{on}^i : i 番目(始点を 0 番目とする)のバス停の乗車需要

d_{off}^i : i 番目のバス停の降車需要

である。

経由するバス停数は 7, 9, 11, 13, 15 の 5 通りで、この停車バス停数ごとに D の高い上位 3 路線、合計 15 本の路線を路線候補とする。GA の処理手順を以下に示す。

Step1. バス停順をランダムに持つ初期個体を生成する。

- Step2. 全固体のDを算出する.
 Step3. 停車バス停数ごとにDが高い3個体(計15個体)が選択され, その中からランダムで2個体を選択し, 交叉・突然変異処理を行い次世代個体を生成する.
 Step4. Step3を繰り返し, 個体が一定数生成されたらStep2に戻る. これを一定世代数に達するまで繰り返し, 最終世代で見つかった上位15個体を路線候補とする.

今回の実験では, 個体数500, 世代数500, 交叉確率0.8, 突然変異確率0.1とした. 交叉・突然変異処理は稲垣⁹⁾の提案した手法を用いる. GAによって求められた路線候補をTable3に示す.

Table 3 : Candidate routes found by the first GA.

Route num.	Depot															D
1	0	1	5	6	7	11	15									613
2	0	1	2	3	7	11	15									610
3	0	1	5	9	13	14	15									583
4	0	1	5	6	2	3	7	11	15							1077
5	0	4	5	1	2	3	7	11	15							1074
6	0	1	5	4	8	9	13	14	15							1069
7	0	1	2	3	7	6	5	9	13	14	15					1891
8	0	1	2	3	7	6	5	9	10	11	15					1863
9	0	1	5	6	7	11	10	9	13	14	15					1806
10	0	1	2	3	7	6	5	4	8	9	13	14	15			2773
11	0	1	2	3	7	11	10	6	5	9	13	14	15			2722
12	0	1	2	3	7	6	5	9	8	12	13	14	15			2708
13	0	1	2	3	7	6	5	4	8	12	13	9	10	11	15	4003
14	0	1	2	3	7	6	5	4	8	12	13	9	10	14	15	3857
15	0	1	2	3	7	11	10	6	5	4	8	9	13	14	15	3781

3.2. 予定所要時間の割り当て

予定時間と実際にかかった時間のズレを計算するために, 路線候補に各バス停間の予定所要時間を割り当てる. 予定所要時間は時刻表に記載される時間のため, 整数の値にする必要がある. また, 多様性を持たせるため, 1つの路線候補に対し複数の予定所要時間を割り当てる. 今回は4パターンの予定所要時間を用いることにする.

制限速度 S , 混雑度 x の道路の速度の期待値 $E_S(x)$ は以下により求められる.

$$E_S(x) = \frac{\alpha_S(x)\{smax_S(x) - 3\}}{\alpha_S(x) + \beta_S(x)} + 3 \quad (12)$$

$E_S(x)$ を切り上げた値を予定所要時間とする.

混雑度 x は(1)式より求めるが, 交通量 V に以下の値を用いることにより, 4パターンの予定所要時間を算出する.

1. 予想交通量
2. 予想交通量 × 2
3. 予想交通量 × 3
4. $\frac{\text{予想交通量}}{2}$

各路線候補に予定所要時間を割り当てた結果をTable4に示す. 1列目は路線候補番号, 2列目は交通量を上記の4パターンで変化させた時の番号であ

Table 4 : Estimated times between depots.

Route num.	Congestion pattern num.	Estimated times														
		Depot order														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	-	1	2	1	2	1	1								
	2	-	1	2	1	2	1	2								
	3	-	1	2	2	2	1	2								
	4	-	1	1	1	1	1	1								
2	1	-	1	1	1	2	1	1								
	2	-	1	2	1	2	1	2								
	3	-	1	2	1	2	1	2								
	4	-	1	1	1	1	1	1								
3	1	-	1	2	2	1	1	1								
	2	-	1	2	2	2	1	1								
	3	-	1	2	2	2	1	1								
	4	-	1	1	1	1	1	1								
4	1	-	1	2	1	1	1	2	1	1						
	2	-	1	2	1	1	1	2	1	2						
	3	-	1	2	2	1	1	2	1	2						
	4	-	1	1	1	1	1	1	1	1						
5	1	-	2	1	2	1	1	2	1	1						
	2	-	2	1	2	2	1	2	1	2						
	3	-	2	1	2	2	1	2	1	2						
	4	-	1	1	1	1	1	1	1	1						
6	1	-	1	2	1	1	2	1	1	1						
	2	-	1	2	1	1	2	2	1	1						
	3	-	1	2	1	1	2	2	1	1						
	4	-	1	1	1	1	1	1	1	1						
7	1	-	1	1	1	2	2	1	2	1	1	1				
	2	-	1	2	1	2	2	1	2	2	1	1				
	3	-	1	2	1	2	2	2	2	1	1					
	4	-	1	1	1	1	1	1	1	1	1	1				
8	1	-	1	1	1	2	2	1	2	1	2	1				
	2	-	1	2	1	2	2	1	2	1	2	2				
	3	-	1	2	1	2	2	2	1	2	2	2				
	4	-	1	1	1	1	1	1	1	1	1	2	1			
9	1	-	1	2	1	2	1	2	1	1	1	1				
	2	-	1	2	1	2	1	2	1	2	1	1				
	3	-	1	2	2	2	1	2	1	2	1	1				
	4	-	1	1	1	1	1	2	1	1	1	1				
10	1	-	1	1	1	2	2	1	1	1	2	1	1	1		
	2	-	1	2	1	2	2	1	1	1	2	2	1	1		
	3	-	1	2	1	2	2	2	1	1	2	2	1	1		
	4	-	1	1	1	1	1	1	1	1	1	1	1	1		
11	1	-	1	1	1	2	1	2	1	1	2	1	1	1		
	2	-	1	2	1	2	1	2	1	1	2	2	1	1		
	3	-	1	2	1	2	1	2	1	2	2	2	1	1		
	4	-	1	1	1	1	1	2	1	1	1	1	1	1		
12	1	-	1	1	1	2	2	1	2	2	2	2	1	1		
	2	-	1	2	1	2	2	1	2	2	3	2	1	1		
	3	-	1	2	1	2	2	2	2	2	6	2	1	1		
	4	-	1	1	1	1	1	1	1	1	1	2	1	1		
13	1	-	1	1	1	2	2	1	1	1	2	2	1	1	2	1
	2	-	1	2	1	2	2	1	1	1	3	2	2	1	2	2
	3	-	1	2	1	2	2	2	1	1	6	2	2	1	2	2
	4	-	1	1	1	1	1	1	1	1	1	2	1	1	2	1
14	1	-	1	1	1	2	2	1	1	1	2	2	1	1	1	1
	2	-	1	2	1	2	2	1	1	1	3	2	2	1	2	1
	3	-	1	2	1	2	2	2	1	1	6	2	2	1	2	1
	4	-	1	1	1	1	1	1	1	1	1	2	1	1	1	1
15	1	-	1	1	1	2	1	2	1	1	1	1	2	1	1	1
	2	-	1	2	1	2	1	2	1	1	1	1	2	2	1	1
	3	-	1	2	1	2	1	2	1	2	1	1	2	2	1	1
	4	-	1	1	1	1	1	2	1	1	1	1	1	1	1	1

る. 3列目以降は始点から順にバス停間の予定所要時間を表して, 交通量が高くなる3番目のパターンほど時間がかかるようになっている.

4. バスダイヤの生成

2回目のGAによる処理で, バスの出発順序, 出発時間を組み合わせ, バスダイヤの生成を行う. バスが始点を出発する間隔を3分, 最終便の発車時刻を57分, 最大車両数20台のバスダイヤを想定し処理を行う.

4.1. 個体の記述

遺伝子にはバスが走行する路線候補番号を持たせ、0番目から順に0分、3分、6分...とバスが始点を出発する順番を表す。また、この路線候補番号に対応させる形で予定所要時間のパターン番号も持たせ、それに応じた時間計算を行う。個体長は20で、1つの個体でバスダイヤを表す。個体の例をTable 5に示す。1行目は使用する路線候補の番号、2行目はその予定所要時間のパターン番号を表している。この個体は、

0分に路線候補5のバス(予定所要時間パターン1)、
3分に路線候補6のバス(予定所要時間パターン0)、
9分に路線候補9のバス(予定所要時間パターン2)、
:
57分に路線候補4のバス(予定所要時間パターン2)

が始点を出発するというバスダイヤになる。バスが出発しない時間もあるが、0分には必ず出発するようにしている。また、使用車両数の閾値を10台に設定し、車両数が10台以下となる個体は無効とする。

Table 5 : Example of individual.

5	6	-	9	0	12	-	-	6	-	5	2	1	-	0	-	-	3	-	4
1	0	-	2	0	1	-	-	0	-	0	1	3	-	1	-	-	0	-	2

4.2. バス運行シミュレーションの設定

バスダイヤの評価のために運行シミュレーションを行う。バスの定員は50人に設定する。また、客の乗り降りにかかる時間は、宇佐美ら⁵⁾による研究を参考にした値を用いる。バスカードは現在は廃止されているためそのデータを削除すると、一人あたりの平均乗車時間は2.93秒となる。平均支払い時間は定額制で1.15秒、距離制で1.94秒となるが、両者をまとめ1.67秒とした。平均降車時間はそのまま2.78秒を用いる。

個体1つのシミュレーションの処理手順を以下に示す。

- Step1. 使用する全てのバスについて、経由するバス停ごとに、バス停番号 d とその予定到着時刻 t_{est} 、車両番号 v (何番目に始点を出発するバスか)をスケジュールとして配列に記録。
- Step2. スケジュールを t_{est} の早い順にソートし、最初のスケジュールから順にループ処理を行う。ここで、

- v 処理中のスケジュールで用いられている車両
- d その車両が停車しているバス停番号
- p^v v に乗っている人数
- p^d d で待っている人数
- p_t 合計輸送人数
- t_{total} スケジュール全体を通しての通算所要時間
- t^v v の通算所要時間
- t_{est} d に到着する予定時刻
- t_{last}^d 最後に d から車両が出発した時間

- t_{delay} 予定よりも遅れた時間
- t_{wait} 出発が予定よりも早くなり、予定時刻まで待機した時間
- t_{leave} 積み残された客が、次のバスに乗るまで待った時間
- t_{on} 一人が乗車にかかる時間= 2.93
- t_{pay} 一人が支払いにかかる時間= 1.67
- t_{off} 一人が降車にかかる時間= 2.78

とする。時間の単位は秒である。

- Step3. $d > 0$ かつ d で降りたい人が n 人($n > 0$)いるなら、

$$t_{delay} += t_{pay}$$

$$t_{delay} += t_{off} \cdot n$$

$$p_t += n$$

さらに、到着が予定より遅い($t^v > t_{est}$)なら、

$$t_{delay} += (t^v - t_{est})n$$

- Step4. $d = 15$ なら Step3 に戻って次のスケジュールへ。

- Step5. $t_{last}^d > 0$ なら、

$$t_{leave} += (t_{total} - t_{last}^d)p^d$$

- Step6. v が次に停まるバス停までの道路の交通量を、

$$\text{予想交通量} \pm \frac{\text{予想交通量}}{2}$$

の範囲の乱数で求める。そこから混雑度を算出し、所要時間をベータ分布乱数から得た速度より求め、 t_{total} と t^v に加算する。

- Step7. v に乗りたい人数を、0~需要表の値までの乱数で求め、 p^d に加算する。

- Step8. $p^d = 0$ になるか、 p^v が定員に達するまで、 $p^d --$ 、 $p^v ++$ 、 $t_{delay} += t_{on}$

- Step9. 出発が予定時刻より遅い($t^v > t_{est}$)なら、

$$t_{delay} += (t^v - t_{est})p^v$$

出発が予定時刻より早い($t^v < t_{est}$)なら、

$$t_{wait} += (t_{est} - t^v)p^v$$

$$t_{total} += t_{est} - t^v$$

$$t^v += t_{est} - t^v$$

- Step10. $p^d > 0$ なら、 $t_{last}^d = t_{total}$

$$p^d = 0 \text{ なら、 } t_{last}^d = 0$$

- Step11. Step3 に戻って次のスケジュールへ。最後のスケジュールまで処理したら終了。

以下の式より、このシミュレーション中に乗客一人あたりが被った超過時間 t_{over} を算出する。

$$t_{over} = \frac{t_{delay} + t_{wait} + t_{leave}}{p_t} \quad (13)$$

4.3. 個体の評価

評価値には t_{over} と p_t を用いる。ある世代の全個体のシミュレーションを行い、評価値をプロットした結果をFig. 7に示す。この時、赤線で示したようなパレートフロントが存在し、 t_{over} と p_t のトレードオフの関係が確認できる。

評価値はシミュレーションのたびに変動するため、個体の優劣を付けるのが難しい。そこで本研究では、モンテカルロ法を用いてシミュレーションを繰り返すし、パレート解となった回数をカウントし、カウン

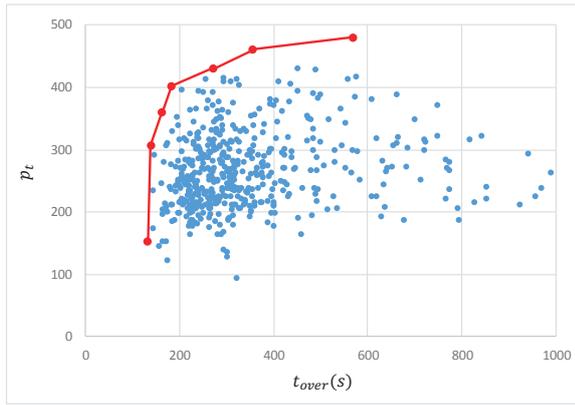


Fig. 7 : Scatter plot which the vertical axis is for p_t and the horizontal for t_{over} .

トの高さで個体の優劣を決定する。パレート解の中で最も p_t が少ない解を (x_1, y_1) , 最も多い解を (x_2, y_2) とする時, 点 $X = (x_1, y_2)$ との距離が近い解ほど優秀と見なす。最も点 X との距離が近いパレート解となった個体のカウントを +3, 次に距離が近くなった個体を +2 する。それ以外は +1 するが, パレートフロントの中心に位置する s_1 個の個体までを対象とし, それ以外は弱パレート解として無視する。 s_1 は以下の値を用いる。

$$s_1 = \frac{\text{パレート解の数}}{3} \quad (14)$$

また, 点 X との距離が遠い s_2 個の個体のカウントを -10 する。 s_2 は以下の値を用いる。

$$s_2 = \frac{\text{パレート解以外の解数}}{3} \quad (15)$$

使用車両数(10~20 台)ごとにカウント数の上位 5 個体を保存する。

4.4. 交叉・突然変異

前節で評価し保存した個体から, 親個体を 2 体決定する。各車両数において, 1 位の個体から順に $\frac{5}{15}, \frac{4}{15}, \frac{3}{15}, \frac{2}{15}, \frac{1}{15}$ の確率で親に選択される。2 体の親で交叉処理を行い, 次世代個体を 2 体生成する。交叉は二点交叉を用いる。交叉をしない場合は, 上記の確率に従って選択された 1 個体をそのままコピーする。

この後に突然変異処理を行う。遺伝子の一箇所をランダムに決定し, その箇所の値を, 別の路線候補番号と予定所要時間パターン番号に変えるか, 削除する。

もし使用車両数が閾値を下回った場合, その個体は破棄する。交叉処理と突然変異処理が終わったら親選択に戻り, 再度処理を行う。これを次世代個体が一定数生成されるまで続ける。

その後, 再び 4.2 節からの処理に戻り, これを一定世代数に達するまで続ける。一定世代数まで達したら, 最後に今まで保存しておいた上位個体を用いて再度モンテカルロシミュレーションを行い, カウ

ント数の高くなった順に最適なバスダイヤとして提示する。

5. 結果

5.1. 提案手法

今回の処理では, 個体数 500, 世代数 500, 交叉確率 0.8, 突然変異確率 0.1, モンテカルロシミュレーション回数 1000 とした。最終的に求められた, 使用車両数ごとのカウント数が最も高くなったバスダイヤを Table 6 に示す。

Table 6 : Best bus timetables of each number of vehicle

Number of vehicle	Pareto count	Route	Individual values															
			14	4	3	2	5	3	5	5			13					
10	806	Route	0	1	1	0	0	2	1	2								
		Pattern										3						
11	565	Route	2	6	10	9	0	0	2	13	4						7	
		Pattern	2	2	1	2	0	0	0	0	0						2	
12	614	Route	12	10	1	12	11	10	10	6	3						2	2
		Pattern	0	0	0	0	0	1	0	2	2						2	2
13	752	Route	10	3	5	12	8	12	1	9	2	1	3				5	2
		Pattern	3	1	0	0	2	3	2	1	2	0	2				0	1
14	540	Route	8	2	2	12	1	12	3	2	0	14	1	3			3	11
		Pattern	2	0	1	3	0	1	1	0	2	1	2	2			3	0
15	555	Route	5	2	12	7	0	6	9	10	11	10	1			3	2	2
		Pattern	1	2	3	3	2	0	1	1	2	2	2			0	2	2
16	942	Route	9	0	6	4	6	9	13	12	0	11	9	3	13	3		8
		Pattern	0	1	0	0	2	2	3	0	0	1	0	2	0	1		0
17	2023	Route	3	5	14	2	4	9	2	5	10	2	5	1	3	14	1	13
		Pattern	0	1	2	2	1	3	2	2	1	1	0	2	3	2	2	0
18	1316	Route	14	1	10	5	10	10	13	6	14	8	6	0	14	2	7	6
		Pattern	0	2	1	1	1	2	1	1	1	2	2	1	1	0	2	2
19	1075	Route	11	9	14	2	4	12	3	13	7	5	3	14	7	0	1	0
		Pattern	1	0	3	3	1	2	3	2	0	2	1	2	0	3	2	0
20	732	Route	4	9	10	0	10	1	13	4	6	3	10	14	1	7	2	4
		Pattern	2	1	0	1	1	1	3	3	1	1	0	3	2	2	0	3

5.2. 従来の GA との比較

期待値のみで個体の評価を行う従来の GA と, 本論文で提案した不確実性を考慮した GA との比較を行う。各パラメータは, 前節で記した提案手法で用いたものと同じ数値を用いる。

モンテカルロ法により各個体の t_{over} と p_t の期待値を算出し, それを評価値として親個体の選択を行い, 最終的にパレート解の集合として解を得ることになる。例として, 使用車両数 17 の得られた 9 個のパレート解を Table 7 に示す。このパレート解の中から, 提案手法の解(Table 6)に対し, 各車両数の解ごとで

Table 7 : 9 pareto solutions of 17 vehicles found by conventional GA

Pareto num.	t_{over}	p_t
1	180.370276	442.1791
2	200.51563	476.5815
3	210.546972	498.8108
4	214.033186	504.1776
5	220.942386	558.5453
6	243.851009	609.9797
7	263.551289	624.9413
8	364.287502	633.2504
9	556.073179	633.6807

t_{over} と p_t それぞれの期待値が最も近くなった個体同士を比較する。今回は、提案手法において最適解のカウンタ数が高かった2車両数(17,18台)と、低かった2車両数(14,15台)の個体について、それぞれ10000回バス運行シミュレーションを行った結果を比較する。その結果を Fig. 8 に示す。黒色が提案手法で得た解(×印が期待値), 青色が従来手法で得た全てのパレート解(赤丸が期待値)のシミュレーション結果である。期待値だけで見ると、車両数 17, 18 台では、提案手法で得た解が従来手法で得た全パレート解より優れたバスダイヤとなっている。提案手法の方が期待値は良い結果となるが、分散は必ずし

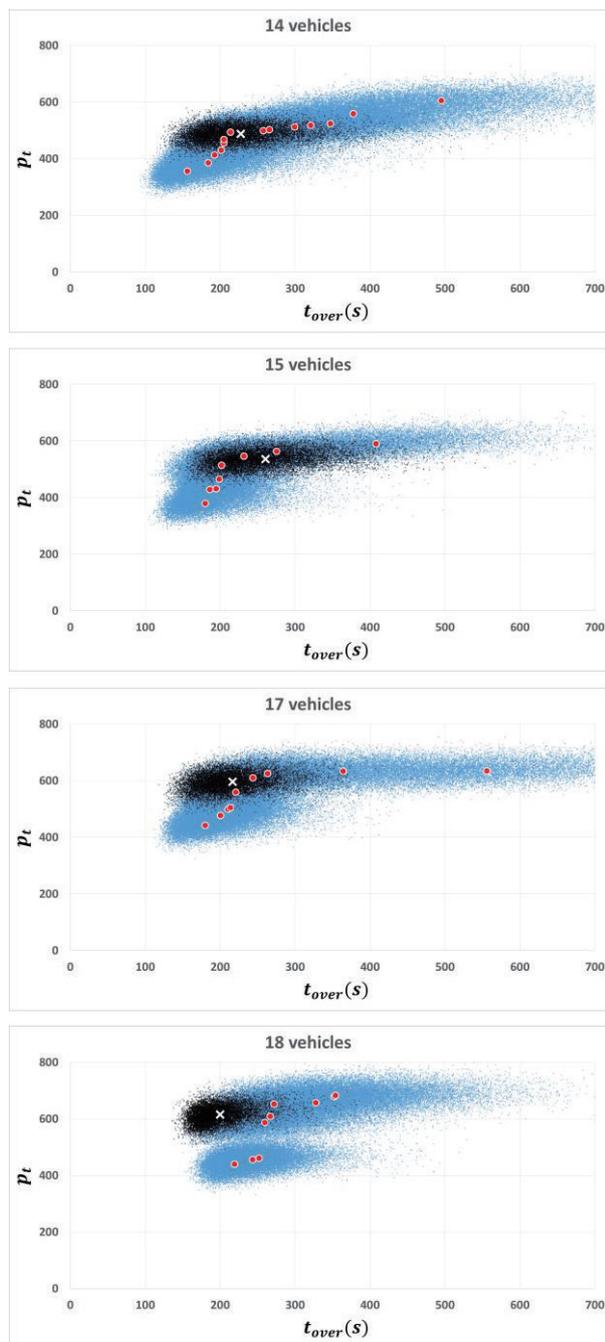


Fig. 8 : Scatter plot of best bus timetables' simulation results.

も低くなるわけではなかった。車両数 14, 15 台では従来手法の方に優れた解があり、カウンタ数が高いほど良い結果となった。

また、従来手法では、求められたバスダイヤはパレート解としての提示になり、そこから選択することになる。それに対し本手法では、カウンタ数が高い順に提示されるため、バスダイヤの優劣が直感的で分かりやすいという利点がある。

6. まとめ

本研究では、道路の混雑状況に合わせたバスダイヤを生成するために、車両速度の予測モデルを生成し、個体集団から分布として評価する GA を適用した。

本手法で生成したバスダイヤの中で、パレート解となったカウンタ数が高いものほど、期待値のみで固体を評価する従来手法で生成した同車両数で運行するバスダイヤよりも、遅延時間と輸送人数の期待値が優れた値になることが分かった。本手法でカウンタ数が低くなった解でもその傾向はあるが、従来手法で得た解より劣る場合もあった。しかし、分布として評価をしても、期待値では勝る個体を得ることができるが、必ずしも分散が低くなるわけではなかった。

今回はカウンタ数の高くなる車両数と低くなる車両数があり、その値によって結果に優劣が現れたが、GA のパラメータや 14, 15 式の値を変更することによって、結果に違いが出るかどうかの検討を行う必要がある。

7. 参考文献

- 1) A. Khani, Y. Shafahi : Transfer optimization in transit networks: Headway and departure time coordination, Proceedings of 14th International IEEE Conference, 1531/1536 (2011)
- 2) 高山 純一, 宮崎 耕輔 : バスダイヤを考慮した最適バス路線網再編計画策定に関する研究, 土木計画学研究・論文集, **13**, 827/836 (1996)
- 3) Y. Wu, J. Tang, Y. Yu, Z. Pan : A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time, Transportation Research Part C: Emerging Technologies, **52**, 15/31 (2015)
- 4) 稲垣 潤 : 経路探索問題における遺伝的アルゴリズムによる解法に関する研究, NDL デジタルコレクション, [online]<http://dl.ndl.go.jp/info:ndljp/pid/3182361> (2001)
- 5) 宇佐美 誠史, 元田 良孝, 金澤 崇 : バス乗降時間の要因に関する基礎研究, 交通工学研究発表会論文報告集, **25**, 269/272 (2005)

構造適応型 Deep Belief Network による時系列データの分析

○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 丸山 罌 (県立広島大学経営情報学部経営情報学科)

Analysis of Time Series Data set by Adaptive Learning Method of Deep Belief Network

*S. Kamada (Dept. of Intelligent Systems, Graduate School of Information Sciences,
Hiroshima City University)

T. Ichimura and R. Maruyama (Faculty of Management and Information Systems,
Prefectural University of Hiroshima)

Abstract— Deep Learning has the hierarchical network architecture to represent complicated features of input patterns. Such architecture is well known to represent higher learning capability compared with conventional models if the best parameters in the optimal network structure are found. We have been developing the adaptive learning method that can discover the optimal network structure. The learning method realizes to construct the network structure with the number of hidden neurons and layers during the learning time. The network structure is self-organized according to given input patterns of big data set. The method records the highest scores in the benchmark test in terms of the stability as well as the classification. In this paper, we develop the recurrent neural network based Deep Belief Network (DBN) to make a higher predictor to the time series data set. We report the experimental result to the Nottingham benchmark data set.

Key Words: Deep Learning, DBN, Adaptive Learning Method, Recurrent Neural Network, Time-series data set

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。例えば、インダストリー4.0における工場の計測データ¹⁾や医療検査データ²⁾などのビックデータをクラウドサーバ上で自動収集し、膨大に蓄積されたデータからその背後にある特徴を抽出し、人間では実現できなかった機能をもつ実用的なシステムを開発するために、人工知能の手法を用いている。ここでは、入力データの瞬間的な判別のために高い分類精度をもつものだけでなく、時系列データを分析し、 t 時刻後の状態を予測する機能の開発が求められている。このような中、Deep Learningは、それまでの機械学習の手法とは異なり、複数のネットワークの階層化と事前学習の機能を実現したことで、高度な分類能力をもつ人工知能として知られるようになり、特に、画像認識の分野で優れた有効性を示している³⁾。

Deep Learningとは、Hintonが提唱した階層型ニューラルネットワークにおける学習手法である⁴⁾。その最大の特徴はネットワーク全体を学習するために行う段階的な事前学習にあると言われている。これは、各層ごとに逐次教師なし学習を行うことで入力データが持つ分布(事前知識)を事前に学習する手法である。事前学習した層を階層的に組み合わせることで、上位層に進むにつれて入力データの特徴を表す抽象的概念を獲得可能である。畳み込みニューラルネットワーク(Convolutional Neural Network; CNN)⁵⁾に見られるように、近年のDeep Learningにおける手法は非常に大規模で複雑なネットワーク構造を持ち、高い学習能力を有している。その問題点の一つは、入力データの分布に応じて最適なネットワーク構造やパラメータ設定を見つけるのが困難であることで、多くの場合、試行錯誤

的に求める必要がある。我々の研究では、このような問題を解決するために、尤度の概念を持つRestricted Boltzmann Machine(RBM)⁶⁾およびRBMを階層化したDeep Belief Network(DBN)⁷⁾に着目し、入力データ空間に応じて適切な大きさのネットワーク構造を学習中に自動で求める構造適応型学習手法を提案した^{8, 9)}。提案手法は、画像認識において既存の手法よりも高い識別性能を示している¹⁰⁾。一方で、近年のIoTにおけるビックデータ分析では時系列データが用いられている。しかしながら、我々がこれまでに開発した提案した構造適応型DBNには時系列データを学習する仕組みはない。このため、時系列データを学習可能な構造適応型DBNの開発が求められている。時系列データを学習できるモデルとしては、リカレントニューラルネットワークの考えをRBMに取り入れたRNN-RBM¹¹⁾が提案されている。本論文では、RNN-RBMの学習中に最適な隠れニューロン数を自動で決定するニューロン生成/消滅アルゴリズムを開発した。さらに、最適な隠れニューロン数に基づいて学習した後、2層以降においても同様に最適なニューロン数を求めながら階層化学習を行うリカレント構造適応型DBNを開発した。時系列ベンチマークデータセットを用いて提案手法の性能を評価したところ、従来の手法と比べて高い性能が得られたため、ここに報告する。

2 構造適応型 Deep Belief Network

2.1 Restricted Boltzmann Machine

RBM⁶⁾は、Fig. 1のように可視層と隠れ層の2層から構成されるネットワーク構造を持ち、確率分布に基づいて入力データに含まれる特徴を隠れニューロン上で学習することできる。

RBMの学習では、与えられた入力データに対して、

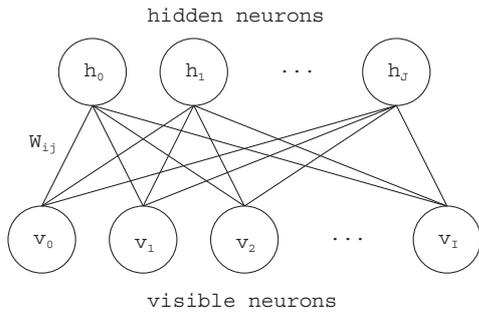


Fig. 1: Structure of RBM

式 (1) のエネルギー関数を最小にするパラメタ $\theta = \{\mathbf{b}, \mathbf{c}, \mathbf{W}\}$ が最尤推定により求められる。

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

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

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

ここで、 v_i は i 番目の可視ニューロン、 h_j は j 番目の隠れニューロンである。 b_i は v_i に対するパラメタ、 c_j は h_j に対するパラメタ、 W_{ij} は v_i と h_j 間の重みである。 $\mathbf{v} \in \{0, 1\}^I$ と $\mathbf{h} \in \{0, 1\}^J$ はそれぞれ可視層と隠れ層のベクトルであり、 I と J は可視ニューロンの数、隠れニューロンの数である。 式 (2) は入力 \mathbf{v} と \mathbf{h} の尤度である。 式 (3) の Z は分配関数 (partition function) であり、 \mathbf{v} と \mathbf{h} の全ての可能な組み合わせに対するエネルギーを示す。 一般的に、RBM の学習では計算量の観点から Contrastive Divergence (CD) 法¹²⁾ によるサンプリングが行われる。

2.2 ニューロン生成/消滅アルゴリズム⁸⁾

我々は、階層型ニューラルネットワークにおける文献¹³⁾ の手法に基づき、RBM の学習中において最適な隠れニューロン数を自動で求めるニューロン生成/消滅アルゴリズムによる構造適応型学習手法を提案している⁸⁾。ニューロン生成は、学習中におけるパラメタの変分に基づいて行われる。例えば、学習データに対して十分な数の隠れニューロンがあれば、一般的に学習が進むにつれて重みベクトルの変分は Fig. 2 のように収束する。一方で、パラメタの変分が収束しない場合は、学習データに対するネットワークの表現能力が不足しており、データが持つ特徴 (確率分布) を隠れニューロン上で線形分離できていないことが原因だと考えられる。このような場合、関連する位置に十分な数の隠れニューロンを挿入することで、多くの次元でデータを表現できると考えられ、ネットワークが安定し、収束する。我々の調査では、RBM の3つのパラメタ $\theta = \{\mathbf{b}, \mathbf{c}, \mathbf{W}\}$ のうち、 \mathbf{c} と \mathbf{W} が CD 法による学習の収束に関係していることを発見し¹⁴⁾、ニューロン生成の条件を式 (4) に示すように定義した。

$$(\alpha_c \cdot dc_j) \cdot (\alpha_w \cdot dW_{ij}) > \theta_G, \quad (4)$$

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変分、 i, j 番目の重みの変分を示し、

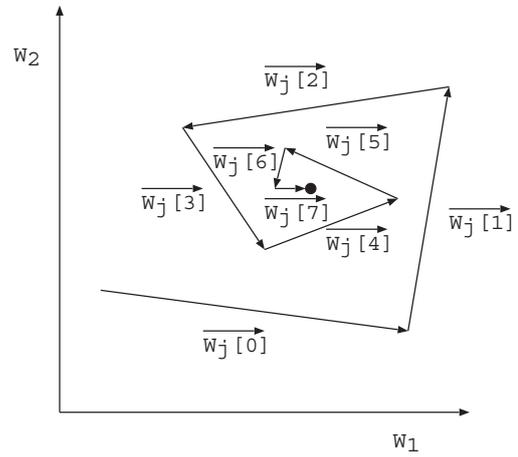


Fig. 2: Convergence situation of a weight vector

α_c 、 α_w は、各パラメタの変分のスケールを調整するために使われるパラメタであり、 θ_G は閾値である。学習中に式 (4) を満たす隠れニューロンが存在する場合、Fig. 3(a) のように、該当する隠れニューロンの近傍に新しいニューロンがその属性 (パラメタや重みの値) を継承するように挿入される。

一方、入力データに対して十分な数の隠れニューロンが生成された後、その中に出力に寄与していない、すなわち冗長な値を出力する隠れニューロンが存在する場合がある。このようなニューロンがある場合、学習における計算コストが増加するだけでなく、ネットワークの構造が複雑になり、誤った出力を生じる場合がある。提案する構造適応型 RBM では、一定の学習の後、式 (5) を満たす隠れニューロンが存在すれば、ニューロン消滅アルゴリズムを適用し、該当する隠れニューロンを Fig. 3(b) のように消去する。式 (5) では、すべての入力データに対する隠れニューロンの出力値の平均をモニタリングし、その値が一定の閾値以下のニューロンを冗長なニューロンとみなし、削除している。

$$\frac{1}{N} \sum_{n=1}^N p(h_j = 1 | \mathbf{v}_n) < \theta_A, \quad (5)$$

$$p(h_j = 1 | \mathbf{v}_n) = \sigma(b_j + \sum_i W_{ij} v_i), \quad (6)$$

ここで、 \mathbf{v}_n は N 個の入力データセットに含まれるある入力データである。 $p(h_j = 1 | \mathbf{v}_n)$ は RBM の可視層に入力データ \mathbf{v}_n が与えられたときの隠れニューロン $h_j \in \{0, 1\}$ の発火確率を示している。ここで、 $\sigma()$ はシグモイド関数で $[0, 1]$ を出力する関数である。 θ_A はしきい値である。

2.3 忘却機能を持った構造学習¹⁵⁾

RBM によるニューロン生成/消滅アルゴリズムにより、入力データ空間に応じて最適なネットワーク構造を学習中に求めることができるが、学習後のネットワーク構造はブラックボックスである。従って、ネットワークに蓄積された重みや隠れニューロンの出力値と、入力データとの関係性が不明確であり、そこから IF-THEN ルールで表現されるような明示的な知識の獲得は困難である。このような問題を解決するため、我々は石川の忘却付き構造学習法¹⁶⁾ に基づいて、RBM の

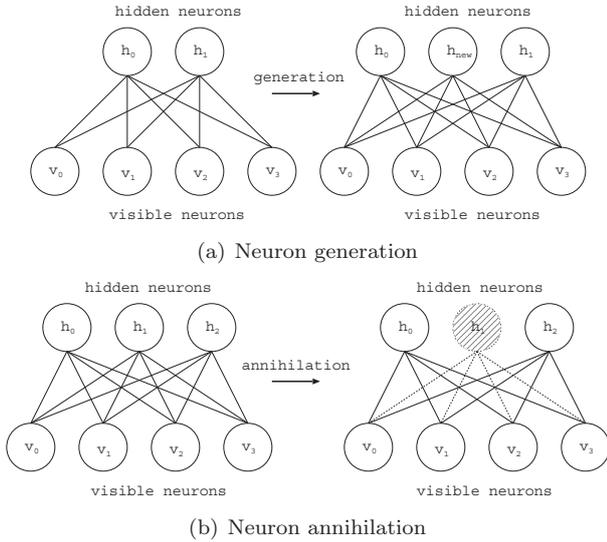


Fig. 3: Adaptive Learning method of RBM

学習後の重みや隠れニューロンの出力活性化値をスパースにする忘却学習手法を開発し、文献¹⁷⁾においてその有効性を示している。式(7)から式(9)は、忘却学習による3種類の目的関数を示している。

$$J_f = J + \epsilon_1 \|\mathbf{W}\|, \quad (7)$$

$$J_h = J + \epsilon_2 \sum_i \min\{1 - h_i, h_i\}, \quad (8)$$

$$J_s = J - \epsilon_3 \|\mathbf{W}'\|, \quad W'_{ij} = \begin{cases} W_{ij}, & \text{if } |W_{ij}| < \theta \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

ここで、 J は学習における通常の目的関数であり、RBMの場合はCD法の推定により生じる誤差を示している。 ϵ_1 、 ϵ_2 、 ϵ_3 は式(7)、式(8)、式(9)の右辺の第2項に対する係数を示している。式(7)では、重みのノルムを目的関数に加えている。これは、重みが必要以上に大きくなることを抑制している。式(8)では、隠れニューロンにおける出力を強制的に2値化するように評価する項を追加している。式(9)では、式(7)の忘却学習によって目的関数の値が通常より大きくなってしまふことを避けるために、しきい値の範囲内の重みに限定して忘却学習を行っている。これらの式を各パラメータに対して偏微分することで学習におけるパラメータの修正量を求めることができる。

2.4 層の生成条件⁹⁾

Deep Belief Network(DBN)は、Hintonが提案したDeep Learningの手法であり、事前学習を行ったRBMを複数積み重ねることで、入力データの特徴を階層的に学習することができる⁷⁾。Fig. 4は、3つのRBMを持つDBNのネットワーク構造を示している。入力層を0層とした際、 $l(> 0)$ 層のエネルギーは式(10)のように定義される。また、 l 層の j 番目の隠れニューロンの発火確率は式(11)のように定義される。

$$E^l = E(\mathbf{h}^{l-1}, \mathbf{h}^l) = -(\mathbf{b}^l)^T \mathbf{h}^{l-1} - (\mathbf{c}^l)^T \mathbf{h}^l - \mathbf{h}^{l-1} \mathbf{W}^l \mathbf{h}^l, \quad (10)$$

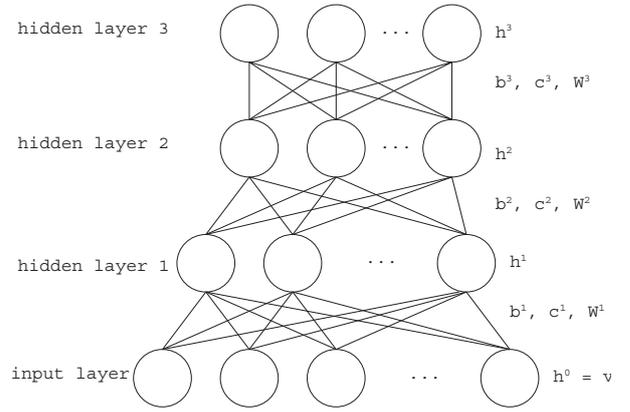


Fig. 4: Hierarchical network structure of DBN

$$p(h_j^l = 1 | \mathbf{h}^{l-1}) = \sigma(\mathbf{c}^l + \mathbf{W}^l \mathbf{h}^{l-1}), \quad (11)$$

ここで、 \mathbf{b}^l 、 \mathbf{c}^l はそれぞれ l 層のRBMの可視ニューロンと隠れニューロンのパラメータであり、 \mathbf{W}^l はその重みである。 $\mathbf{h}^l \in \{0, 1\}$ は隠れニューロンの状態ベクトルである。

我々は、DBNの学習中において、入力データに対して最適な隠れ層の数を自動で求める構造適応型学習手法を開発した¹⁰⁾。構造適応型RBMにおけるニューロン生成アルゴリズムでは、学習中のWD(パラメータ \mathbf{c} と \mathbf{W} の変分)をモニタリングしたが、提案する構造適応型DBNでは、個々のRBMの安定性ではなく、ネットワーク全体の安定性を測るため、ネットワーク全体のWDとエネルギーを式(12)と式(13)で観察した。

$$\sum_{l=1}^k (\alpha_{WD} \cdot WD^l) > \theta_{L1}, \quad (12)$$

$$\sum_{l=1}^k (\alpha_E \cdot E^l) > \theta_{L2}, \quad (13)$$

ここで、 WD^l は l 層のRBMにおける \mathbf{c}^l と \mathbf{W}^l の変分を示している。 E^l はエネルギー関数である。 α_{WD} と α_E はスケールを調整するために使われるパラメータであり、 θ_{L1} 、 θ_{L2} は閾値である。ある k 層のRBMの学習中に式(12)および式(13)を同時に満たす場合、入力データに対するネットワーク全体の表現能力が不足していると考え、 k 層の学習が終了した時点で、 $k+1$ 層のRBMを生成し、学習を行う。ここでもRBMのニューロン生成アルゴリズムと同様に、 $k+1$ 層のRBMのパラメータ \mathbf{b} 、 \mathbf{c} 、 \mathbf{W} の初期値が k 層のRBMから継承される。

3 時系列データの分析

3.1 RNN-RBM

RNN-RBM¹¹⁾はRBMにおいて、リカレントニューラルネットワークを実現した時系列学習モデルである。厳密に言うと、RNN-RBMはRecurrent Temporal RBM¹⁸⁾をもとに、BP学習によるリカレントニューラルネットワークの一種であるBPTT(Backpropagation through time)¹⁹⁾学習をRBMで実現したモデルになっている。RNN-RBMでは、通常のRBMの可視層と隠れ層に加えて、過去の時系列に関する文脈を表現する状

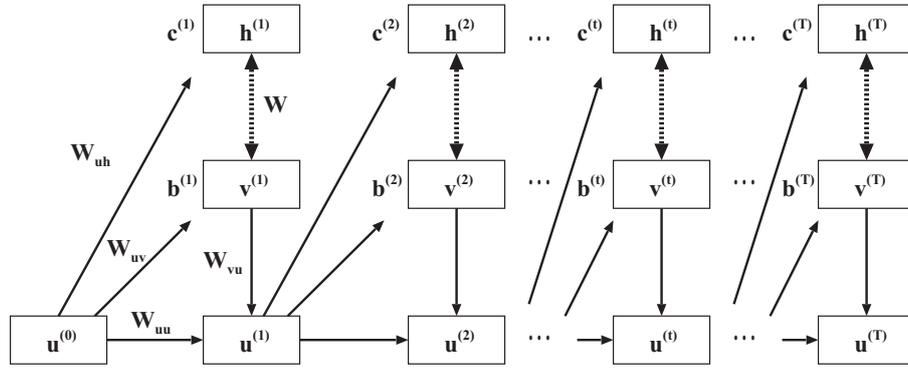


Fig. 5: Structure of RNN-RBM

態 $\mathbf{u} \in \{0, 1\}^K$ を持つ. 長さ T の入力データのシーケンスを $\mathbf{V} = \{v^{(1)}, \dots, v^{(t)}, \dots, v^{(T)}\}$ とした場合, ある時刻 t の入力 $v^{(t)}$ に対する可視層と隠れ層のパラメータ $\mathbf{b}^{(t)}$, $\mathbf{c}^{(t)}$ が, 時刻 $t-1$ の状態 $\mathbf{u}^{(t-1)}$ のみから, 式 (14), 式 (15) によって求められる. 時刻 t の状態 $\mathbf{u}^{(t)}$ は, 時刻 t の入力 $v^{(t)}$ と時刻 $t-1$ の状態 $\mathbf{u}^{(t-1)}$ から式 (16) のように更新される.

$$\mathbf{b}^{(t)} = \mathbf{b} + \mathbf{W}_{uv} \mathbf{u}^{(t-1)} \quad (14)$$

$$\mathbf{c}^{(t)} = \mathbf{c} + \mathbf{W}_{uh} \mathbf{u}^{(t-1)} \quad (15)$$

$$\mathbf{u}^{(t)} = \sigma(\mathbf{u} + \mathbf{W}_{uu} \mathbf{u}^{(t-1)} + \mathbf{W}_{vu} v^{(t)}) \quad (16)$$

ここで, $\sigma()$ はシグモイド関数である. 図 5 は RNN-RBM における計算の流れを示している. $\mathbf{u}^{(0)}$ は状態の初期値で, 任意の値が与えられる. RNN-RBM の学習は, 各時刻 t において計算された $\mathbf{b}^{(t)}$, $\mathbf{c}^{(t)}$ と重み \mathbf{W} を用いて, $v^{(t)}$ と $h^{(t)}$ 間の学習が通常の RBM と同じように行われ (一般的に CD 法), 学習による誤差 (損失) が計算される. 時刻 T までの誤差を計算した後, T から過去に遡るように学習パラメータ $\theta = \{\mathbf{b}, \mathbf{c}, \mathbf{W}, \mathbf{u}, \mathbf{W}_{uv}, \mathbf{W}_{uh}, \mathbf{W}_{vu}, \mathbf{W}_{uu}\}$ の勾配が BPTT により計算され, 更新される.

3.2 構造適応型 RNN-DBN

RNN-RBM は, RBM と比べて, 1) 時刻 t の入力に対するパラメータ $\mathbf{b}^{(t)}$ と $\mathbf{c}^{(t)}$ が過去の時刻 $t-1$ の状態から計算される, 2) 勾配計算に BPTT が使われる, という点において異なっているが, RBM の構造自体が変わりではなく, 通常の RBM と同様に CD 法による学習が行える. すなわち, CD 法では, 与えられた入力データに対して目的関数を最小化するパラメータ \mathbf{b} , \mathbf{c} , \mathbf{W} を学習する. 従って, 学習中のパラメータの変分が大きい場合, 入力データに対する表現能力を補うためニューロン生成を行う手法を取り入れることができ, 2.2 節の構造適応型 RBM におけるニューロン生成/消滅アルゴリズムを RNN-RBM で実現した構造適応型 RNN-DBN を提案する.

構造適応型 RBM と同様に, 学習中における \mathbf{c} , \mathbf{W} をモニタリングし, 式 (4), 式 (5) を満たす場合, ニューロン生成, 消滅を行う RNN-RBM を開発した. さらに, 構造適応型 RNN-RBM による学習において, 学習後の重みや隠れニューロンの出力値をスパースにする忘却学習を実装した. 加えて, 2 層以降においても最適な隠れニューロン数を求めながら階層化を行う構造適応型 RNN-DBN を開発した. RNN-RBM では, 時刻 t の入

力 $v^{(t)}$ に対する隠れニューロン $h^{(t)}$ が時刻 $t-1$ の状態 $\mathbf{u}^{(t-1)}$ から決定的に計算される. つまり, ある入力に対する隠れニューロンの出力値が一意に求まるため, この出力値を次の層における入力とし, 2 層以降においても構造適応型 RNN-RBM による学習を行った.

4 実験

4.1 データセット

提案するリカレント構造適応型 DBN の性能を評価するために, 時系列ベンチマークテスト “Nottingham”²⁰⁾ を用いた. Nottingham は, 88 個の階調で表現される MIDI 形式のピアノの音声データで, 694 個の訓練データと, 170 個のテストデータから構成される. 各データから, 約 210 個の長さの時系列データを取得できる.

リカレント型構造適応型 DBN のパラメータとして, 学習率は 0.01, 学習のバッチサイズは 100 とした. 構造適応型 RBM のパラメータとして, $\theta_G = 0.001$, $\theta_A = 0.2$ 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) 24Core Xeon E5-2670 v3 2.3GHz, GPU = Tesla K80 4992 24GB \times 3, Memory = 64GB, OS = Centos 6.7 64 bit.

4.2 実験結果

Fig. 6 は, 従来の RNN-RBM と提案する構造適応型 RNN-RBM の学習状況を示している. Fig. 6(a) から Fig. 6(e) まで, それぞれ学習の各イタレーションに対するエネルギー, 誤差, \mathbf{W} の変分, \mathbf{c} の変分, 隠れニューロンの数を示している. 従来の RNN-RBM では, Fig. 6(c), Fig. 6(d) に示すように, パラメータの変分が大きく振動している箇所が見られた. このことが原因となって, Fig. 6(a) や Fig. 6(b) で示すように, 従来の RNN-RBM ではエネルギーや誤差が安定しなかったと考えられる. 一方, 構造適応型 RNN-RBM では, 20 回目の epoch までは従来の RNN-RBM と同様にパラメータの変分が振動している部分が見られたが, それ以降の学習では Fig. 6(e) で見られるように, 最適と考えられるニューロン数が生成され, パラメータの変分が徐々に小さくなり収束した. エネルギーや誤差についても同様な結果になった.

Fig. 7 は 1 層目の RNN-RBM において, 忘却学習を適用した場合 (Fig. 7(a)) と適用しなかった場合 (Fig. 7(b)) の重みの分布を示している. 忘却学習により, 全体の分布の偏りが少なくなり, スパースな構造が得られ, その有効性を RNN-RBM においても示すことができた.

Table 1 は, 学習後の精度を示している. 提案手法

Table 1: Accuracy

	No. layers	Error(Training)	Error(Test)	Correct ratio (Test)
Traditional RNN-RBM	1	0.945	1.704	71.7%
Adaptive RNN-RBM	1	0.881	1.240	76.5%
Traditional RNN-DBN	4	0.217	1.381	75.8%
Adaptive RNN-DBN	4	0.101	0.133	85.3%

の構造適応型 RNN-RBM は従来の RNN-RBM も訓練データだけでなく、テストデータに対しても誤差が小さくなり、予測に対する正答率が高くなった。また、従来の RNN-RBM では、階層化を行っても誤差や正答率の精度がそこまで向上しなかった。これは、下位層の事前学習が最適なパラメタに基づいて行われていなかったからだと考えられる。一方、構造適応型 RNN-DBN では、エネルギー、誤差、正答率に関して比較した手法の中で最も良い結果を示すことができた。

5 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。これらは特に画像認識の分野で優れた成果を残しているが、入力データ空間に応じたネットワーク構造の最適化や学習結果の可視化の課題は解決されていない。また、Deep Learning の手法として、畳込みニューラルネットワークがよく使われているが、計算量が多いことと、パラメタの設定が難しいと考えられ、学習モデルの構築には多くの経験が必要である。本研究では、尤度の概念を用いた統計的なモデルであるため、構造の最適性の証明や学習結果の可視化に優れている RBM, DBN に着目し、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 DBN を開発した。ここでは、学習中におけるネットワーク全体のパラメタの変分とエネルギーを観察し、これらの値が一定のしきい値を超えた場合、新しい RBM の層を生成する階層化の条件を提案した。さらに、RBM の時系列学習モデルである RNN-RBM をもとに、リカレント構造適応型 RNN-DBN を開発した。ベンチマークテストを用いた実験を行ったところ、本論文で提案するリカレント構造適応型 DBN は時系列データに対しても高い予測精度を持つことが分かった。

謝辞

本研究開発は総務省 SCOPE(受付番号: 162308002)の委託を受けたものです。

参考文献

- 1) T.Lyons and M.Skitmore: *Project risk management in the Queensland engineering construction industry : a survey*, International Journal of Project Management, Vol.22, No.1, 51/61 (2012)
- 2) M.Suka, T.Ichimura and K.Yoshida: *Development of coronary heart disease databases*, Proc. of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES2004), Vol.2, 1081/1088 (2004)
- 3) V.Le.Quoc, R.Marc's Aurelio, et.al: *Building high-level features using large scale unsupervised learning*, International Conference in Machine Learning (2012)
- 4) Y.Bengio: *Learning Deep Architectures for AI*, Foundations and Trends in Machine Learning archive, Vol.2, No.1, 1/127 (2009)
- 5) Y.LeCun, et.al.: *Backpropagation applied to handwritten zip code recognition*, Neural Computation, Vol.1, No.4, 541/551 (1989)
- 6) 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)
- 7) 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)
- 8) S.Kamada and T.Ichimura: *An Adaptive Learning Method of Restricted Boltzmann Machine by Neuron Generation and Annihilation Algorithm*, Proc. of IEEE SMC 2016, 1273/1278 (2016)
- 9) 鎌田真, 市村匠: **忘却機能を持った Deep Belief Network による構造適応型学習**, 計測自動制御学会第 9 回コンピュータ・インテリジェンス研究会講演論文集, 92/97 (2016)
- 10) S.Kamada and T.Ichimura: *An Adaptive Learning Method of Deep Belief Network by Layer Generation Algorithm*, Proc. of IEEE TENCON2016 (to appear in 2016)
- 11) N.Boulanger-Lewandowski, Y.Bengio, and P.Vincent: *Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription*, Proc. of the 29th International Conference on Machine Learning (ICML2012) (2012)
- 12) G.E.Hinton: *Training products of experts by minimizing contrastive divergence*, Neural Computation, Vol.14, 1771/1800 (2002)
- 13) T.Ichimura and K.Yoshida Eds.: *Knowledge-Based Intelligent Systems for Health Care*, Advanced Knowledge International (ISBN 0-9751004-4-0) (2004)
- 14) 鎌田真, 市村匠, 原章: **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**, 計測自動制御学会第 8 回コンピュータ・インテリジェンス研究会講演論文集, 90/96 (2015)
- 15) S.Kamada and T.Ichimura: *A Structural Learning Method of Restricted Boltzmann Machine by Neuron Generation and Annihilation Algorithm*, Neural Information Processing, Vol.9950 of the series Lecture Notes in Computer Science, 372/380 (2016)
- 16) M.Ishikawa: *Structural Learning with Forgetting*, Neural Networks, Vol.9, No.3, 509-521 (1996)
- 17) S.Kamada, Y.Fujii and T.Ichimura: *Structural Learning Method of Restricted Boltzmann Machine with Forgetting*, Proc. of 17th Annual Meeting of Self-Organizing Maps in Japanese, 13.1/6 (2016)
- 18) I.Sutskever, G.E.Hinton, and G.Taylor: *The Recurrent Temporal Restricted Boltzmann Machine*, Advances in Neural Information Processing Systems, Vol.21 (NIPS 2008)
- 19) D.E.Rumelhart, G.E.Hinton, and R.J.Williams: *Learning internal representations by error propagation*, Parallel distributed processing: explorations in the microstructure of cognition, Vol.1, 318/362, MIT Press (1986)
- 20) Nottingham, <http://www-etud.iro.umontreal.ca/~boulanni/icml2012> (2016 年 11 月 19 日閲覧)

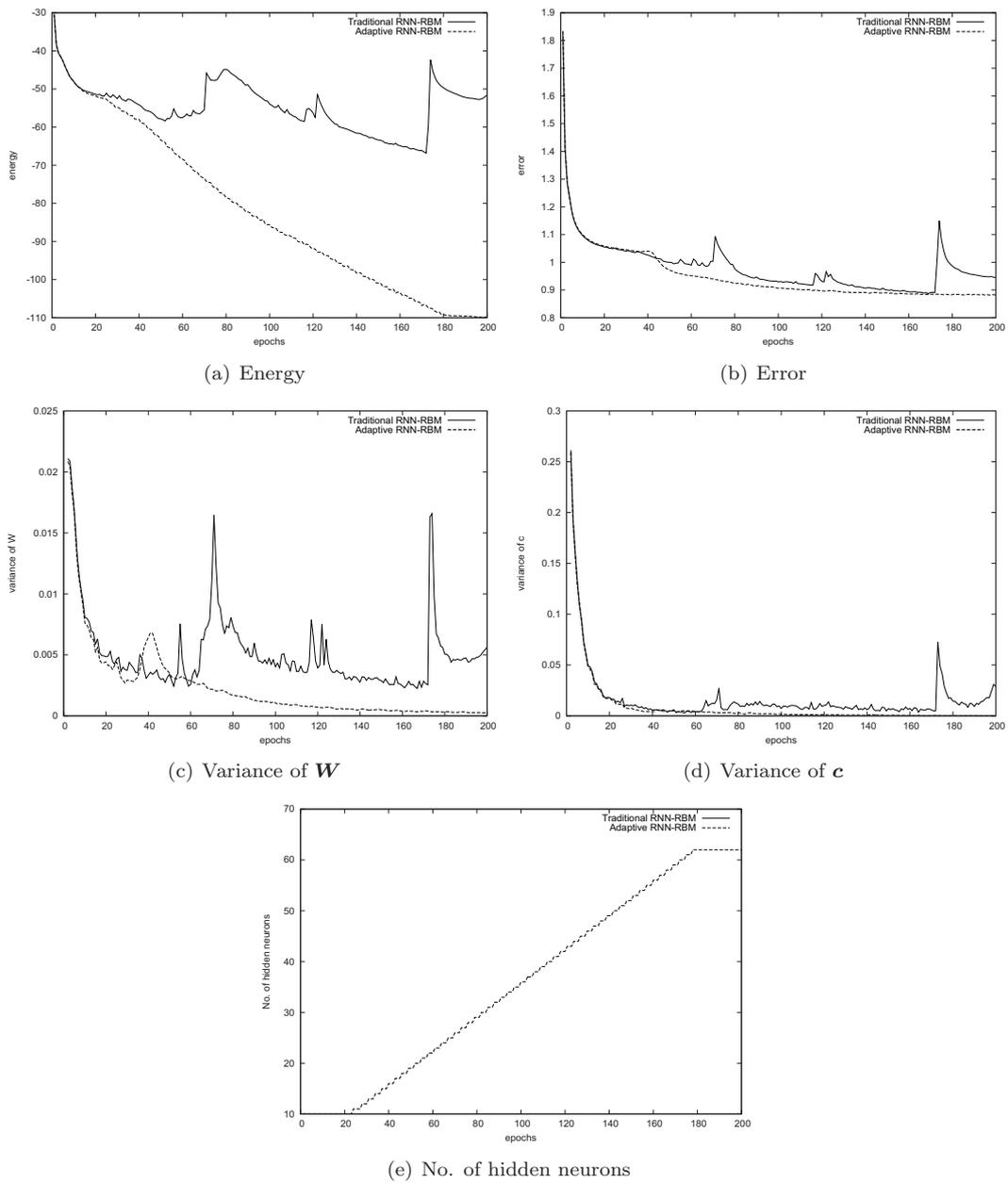


Fig. 6: Experimental Results on Nottingham (layer 1)

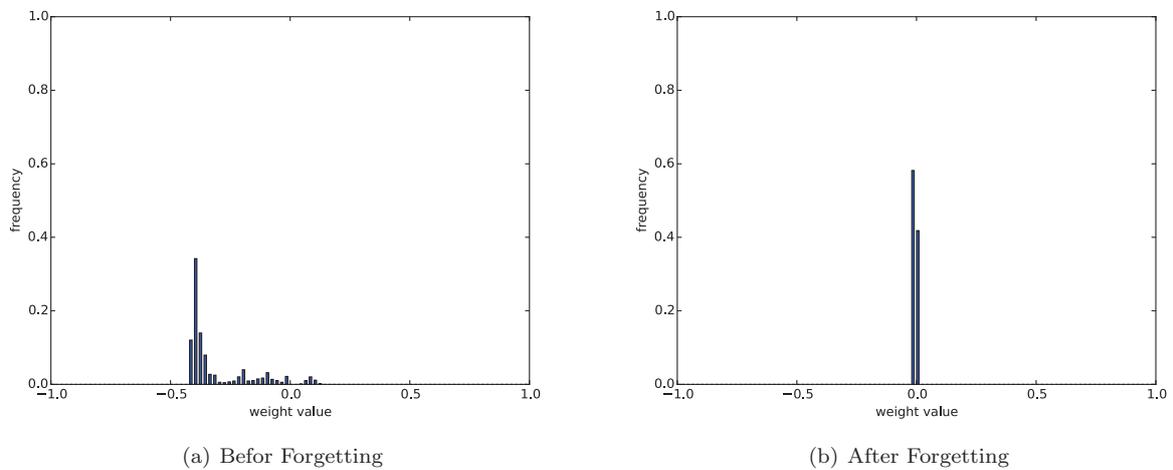


Fig. 7: Weight Distribution on Nottingham (layer 1)

両眼水平速度対による動的衝突予測法

○森文彦 菅野直敏 (玉川大学)

Collision Prediction by Apparent Binocular Horizontal Velocity Pair under the Condition of Ego- and Object Motion

* F. Mori and N. Sugano (Tamagawa University)

Abstract— We propose a new aperture problem-free method of collision prediction that is effective when any object including ego is able to be in either a state of motion or stillness. A new concept named “apparent binocular horizontal velocity pair” is introduced to enable a robot to arrive at a goal smoothly without colliding with a moving or a stationary object. The collision location and time are calculated directly from the binocular horizontal velocity pair. The effectiveness was confirmed by a robot with the online stereo video camera.

Key Words: Binocular velocity pair, Collision prediction, Stereo video image, Ego- and object motion, Robot, Aperture problem

1 はじめに

ここでは、自律移動ロボット、自動運転の車椅子や自動車、 「自己を含むすべてのものが静止あるいは運動の状態にある環境」で目的地まで障害物と衝突しないで移動するための要素技術の1つである視覚センサによる動的衝突予測法を取り扱う。この環境では、自己も移動するため、自己以外のものもすべて複雑な動状態となるので、素早い正確な計測と動的衝突予測が必要となる。ここでいう「動的衝突予測」とは、「現在の動状態がそのまま続いた場合、どの対象物が自己のどこに、何時、衝突するかを各時刻で予測すること」をいう。この機能は、例えば、人間が目的地まで衝突しないで安全に到達するために必須のものである。

以下で視覚を用いた関連する先行研究について述べる。動的衝突予測に関する研究は少ないので、対象物の3次元移動を取り扱っているものも含めて述べる。衝突予測・回避の問題解決の手法には、(1)単眼カメラによるものと(2)ステレオカメラによるものに大別される。前者の代表的なものは、対象物の画像上の位置移動とエピポール(移動体のFOE: focus of expansion)からカメラを含むカメラの光軸に垂直な前額平行面と対象物が衝突する時刻を予測する手法[1]である。これには、各対象物のエピポールの正確な推定が必要である。また、衝突部位の推定には、各対象物のある時刻の3次元的位置など何らかの追加的情報を必要とする。一般に、単眼情報だけでは情報不足になる[2][3][4]。後者のステレオカメラによる手法は、(2-1)時刻 t と $t+1$ における対象物の空間位置とそれらの時刻間の対応付けによる手法と(2-2)時刻 t における対象物の空間位置(両眼カメラの画素の対応付け)と両眼における2つの速度分布による手法に分類される。

(2-1)の試みの1つとして、各時刻の環境の3次元構造を求め、それらを時刻間でうまくつなぎ合わせ3次元運動速度を得る手法がある。このような考え方は、例えば文献[5]で言及され[6]、工学的試みも行われてきている[4][7][8][9]。文献[4][7]の手法はほとんど同じものであり、時刻間の対応は片眼における速度分布に基づいて行われている。現在では、窓問題(局所領域の解析だけでは、境界線が物理的にどの方向に移動したかを知ることは原理的にできないという問題、運動視や速度分布を取り扱う場合の重要問題:付録参照)の生じにくいコーナーなどの特徴点で速度分布を求め、その動きより時刻間の空間の

対応付けを行い対象物の3次元速度を抽出する手法が取られている。この時の処理の多さを仮に、対応点の探索のみで計るとすれば、1つの次元における対応点の探索範囲を n (画素)とすると両眼視差抽出で n 、速度抽出で n^2 の合計(n^2+n)ということになる。この手法の場合、時刻間の対応を画素単位で行うとすれば、例えば、水平視野 70deg (320画素)のカメラ、フレーム間時間 0.1秒、5m先の対象物が秒速 1m で接近する場合水平衝突位置の誤差 $E \sim 100\text{cm}$ のような重大な短所が生じる($E=Z^2 \Delta/D$ 、ただし、 Z は時刻 t における対象物の位置、 D はフレーム間移動距離、 $\Delta = Z \tan 35^\circ / 160$ は、1画素当たりの距離 Z における水平の長さ($Z=5\text{m}$ の上記条件の場合 $\sim 2.2\text{cm}$))。

もう1つの方法(2-2)の1つである文献[3]は、文献[10][16]および[11]の言及する両眼視差 d の時間微分 ($\partial d / \partial t$) と両眼視差との比 ($\partial d / \partial t$) / d を3次元運動速度抽出の基本量として求め、この量と両眼の速度分布の組み合わせで3次元移動速度を抽出している。文献[3]では、窓問題を避けるため、人工的に点群をシーンに投影し、それらの2次元速度分布と両眼視差を求めて自己のみ移動する環境で実験を行っている。また、衝突予測そのものは行っていない。この時の対応点の探索量は、同様に求めると合計($2n^2+n$)となる。

以上から分るように、ほとんどの研究で何らかの形で対象物群の画像上の速度分布が正確に抽出できていることを前提としている。約30年前から速度分布に関する理論的研究が盛んに行われてきた(例えば、文献[12][13][14])。この時の大きな問題点は、速度分布をいかに正確に求めるかということであった。窓問題のため、局所的に速度を求めることは不可能であるので、投影された光点群などで模様付けされたシーンが使用されてきた。

これまで行われてきた速度分布からの自己移動パラメータ抽出、両眼視差と速度分布から移動体の3次元運動の抽出などの研究は、任意の3次元運動を取り扱うことを視野にいれて行われてきたためか、各点の画像上の正確な2次元運動の抽出が必須であった。残念ながら、窓問題のため前述のような特徴点以外ではそれを求めることはできない。そのため、衝突予測に欠かすことのできない視野の水平線近傍の領域にある多数のエッジやそれ以外の視野内に通常の実空間で極めて多く存在する電柱や建物などの重要な直線的エッジは処理の対象から外れてしまっている。これは、研究で扱われる環境が、水平

面上を水平に移動する場合がほとんどであるにもかかわらず任意の3次元運動を扱うという伝統的研究パラダイムを採用してきたことに起因するのかもしれない。

当該研究分野の大きな課題は、窓問題を解決すること、速度抽出のための探索範囲を少なくすること、正確な衝突予測のできる簡単な計算式を見つけることである。この課題の解決を検討した結果、以下の4条件下では、窓問題を考慮しなくてもよいこと及びこの時の探索量の合計は、 $(2-3n)$ であることが分った。(1)ステレオカメラの光軸は、水平面に平行であり、水平に移動する。(2)カメラ以外の移動体も、水平面上を水平に移動する。(3)カメラの光軸と移動方向は、任意の角度で固定されている(複数のステレオカメラを使用すれば全方位の衝突予測が可能である)。(4)frame間では、等速直線運動で近似できる(軌道に沿う運動によって生じる公転的回転運動は良いが、自転的回転は含まない)。一般に、上記条件を備えた環境には、ほとんどの日常生活場面が含まれているので自然なものと思われる。この段階で残されている課題は、「正確な衝突予測のできる簡単な計算式」を求めることである。文献[6][15][16]は、「速度比(v_r/v_l)を水平面上の速度方向抽出に使うこと」を提案していた。残念ながら、この量は、水平面上の速度方向に関連はしているが、速度の大きさを反映しないこと、値が $-\infty$ から $+\infty$ まで変化すること、そのままでは、衝突部位・時刻を特定できないことなどのため1982年当時より、取り扱い難い量であった。最近この問題を解く研究の機会を得て考察を行った結果、第2章で述べる、速度対から直接衝突時刻と位置を予測する新しい衝突予測・回避手法を得ることができた[17][18][19]。また、輝度・色が一樣な領域では、安定した領域分割など何らかの追加処理が無い限り、水平面上での移動速度を抽出できないことはいうまでもない。当該研究で新たに行ったことは、(1)窓問題の生じない条件を設定したこと(任意の方位のエッジで衝突予測が可能なる条件)、(2)簡単かつ精密な衝突時刻と位置の予測式を求めたこと、(3)ステレオカメラを備えた簡単な移動ロボットでその有効性を確かめたことである。ここで設定した窓問題の生じない環境では、前述の(2-1)の手法でも原理的には可能であるが、同じ高さの対応点をsub-pixelで求めることなどの困難さがある。

当該研究は、両眼速度分布に関する文献[3][6][15][16]の延長線上にあるものである。文献[3]との相違について若干補足する。当該研究で述べた実験環境においてではあるが、ここでは、水平以外の任意の傾きのエッジの真の速度分布の抽出法が示されている。文献[3]で提案されている実空間上での相対速度(V_x, V_z)から衝突位置 X_T と時刻 T を算出することも考えられるが、直接初等幾何学で導出した提案式の方が単純で誤差が出難い。もし、(V_x, V_z)が必要であるなら、逆に、当該研究で導出されている画像上の位置情報の不要な(X_T, T)と既に両眼視差から求められている現在の実空間上の位置から計算した方が、用途にも依るが、単純かつ実用的と考えられる。

第2章では、衝突部位と衝突時刻の推定結果例を示す。第3章では、自己移動中に移動対象物と静止対象物の混在する環境で衝突予測と回避行動が実時間で行うことができるかどうかの実験結果を示す。

2 見かけの両眼水平速度対による衝突予測方法

2.1 原理の説明

対象物が水平面上奥行き方向に移動するとその移動方向に応じて右眼カメラ上での動き(δ_R)と左眼カメラ上での動き(δ_L)に相違が生じる。逆に、この相違を計測し、その相違から対象物の衝突部位や衝突時刻を予測しようというのが両眼速度対法である。ここで、対象物の奥行き移動方向と両眼速度対の関係を図1に示す。図1の状況は、対象物の視野内における位置や自己移動方向に関係なく相対空間として成立するものである。図1においてLとRは、左眼カメラと右眼カメラであり、下の太い矢印は、それぞれ左眼カメラ画像上の速度と右眼カメラ画像上の速度(両眼速度対)を示し、点線の矢印は、時刻(t)における対象物の存在方向を示し、カメラを始点とした実線の矢印は、時刻($t + \Delta$)における対象物の存在方向を示す。また、小円は、対象物を示し、小円間の矢印は、対象物の時刻(t)

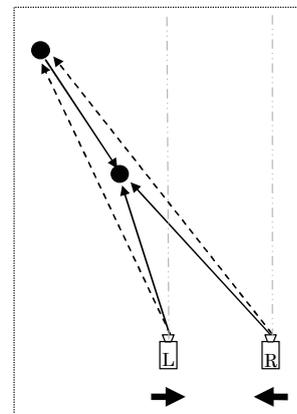


Fig. 1: Relation between apparent binocular horizontal pair and depth movement.

から($t + \Delta$)までの水平面上の移動を示している。

当該手法は、両眼カメラの速度対から窓問題を考慮しなくてもよい環境で、自己のどの部位(X_T)に何時(T)衝突するかを各時刻で予測するというものである。

図2に衝突部位(X_T)と衝突時刻(T)抽出の詳しい原理図を示す。この図2において、高さ方向の座標軸(y軸)が示されていないが、これは「対象物は、水平面上を水平に移動する」という最初に設定した実験条件にゆらいするものである。高さ座標Yが必要なのは、垂直エッジ以外の角度のエッジの場合(式(6)と式(7)参照)である。

原点は、右眼カメラ、x軸は、両眼カメラを結ぶ水平線、z軸は、前方(光軸)である。図中の記号の意味は以下のとおりである。

- P_0 : 対象点の時刻tにおける位置 (X_0, Z_0)
- P : 対象点の時刻 $t + \Delta t$ における位置(X, Z)
- PL: 点Pを通り額面と平行な面と直線LPとの交点
- PR: 点Pを通り額面と平行な面と直線RPとの交点
- α : 一画素あたりの視角
- a: 左右のカメラ間の距離
- δ_L : 点Pの左眼カメラ上の移動画素数: 左眼速度
- δ_R : 点Pの右眼カメラ上の移動画素数: 右眼速度

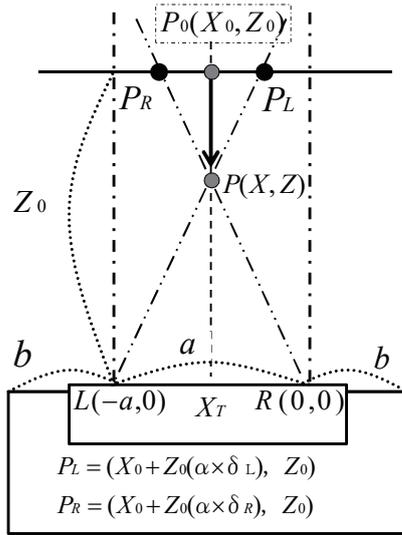


Fig. 2: Principle of collision point (X_T) and time (T) extraction.

両眼速度対 (δ_L, δ_R) から衝突部位 (X_T) と衝突時刻 (T) を求める具体的な計算式を式(1)と式(2)に示す.

$$X_T = \frac{a\delta_R}{\delta_L - \delta_R} \quad \dots (1)$$

$$T = \frac{a}{\alpha Z_0 (\delta_L - \delta_R)} \quad \dots (2)$$

衝突条件は、式(3)と(4)で与えられる.

$$-(a+b) \leq X_T \leq b \quad \dots (3)$$

$$T > 0 \quad \dots (4)$$

式(1)(2)は、正しい水平移動距離が求まっていることを仮定して得たものであるが、以下では、任意に位置の任意の向きのエッジの任意の移動に対するそのシステムの誤差について検討する. エッジの方向と水平線との角度を θ , 真の移動方向と水平線との角度を β (未知), 計測されるエッジの水平移動距離を δ' とすると式(5)が成り立つ (付録参照).

$$\delta = \frac{\delta'}{1 + \frac{\tan \beta}{\tan \theta}} \quad \dots (5)$$

対象点 $P(t)$ の自己への点 P_T への衝突過程を図3に示す. 図3から式(6)が導出されることは明らかである.

$$\tan \beta_R = \frac{Y_0}{X_T}, \quad \tan \beta_L = \frac{Y_0}{X_T + a} \quad \dots (6)$$

式(1)と式(6)から衝突位置 X_T が式(7)として求まる.

$$X_T = \frac{a}{\delta'_L / \delta'_R - 1} + \frac{Y_0}{\tan \theta} \quad \dots (7)$$

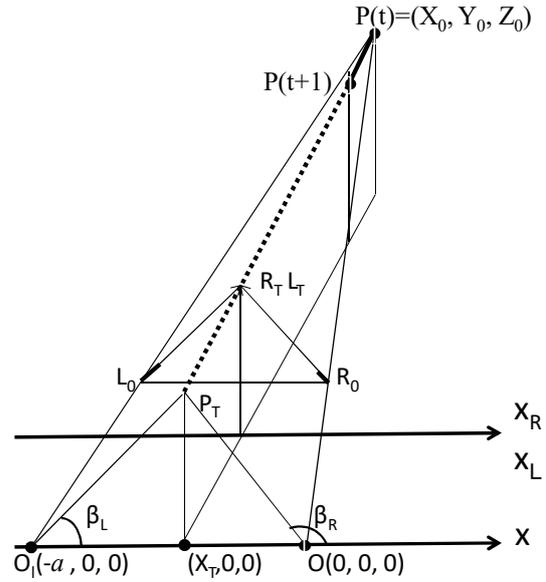


Fig. 3: Collision process of object $P(t)$ to point P_T . (The small circle is a window of the window problem. See an appendix.)

衝突位置の予測式から明らかのように分母と分子の誤差が相殺するので誤差は全く生じない. 従って, 当然の結果として未知であった角度 β が式(6)のように定まる (垂直移動距離も求まる). 従って, 当該手法が窓問題に影響されないものであることが分る. しかしながら, 当該研究では, 実時間衝突予測の実験で有効性を確かめようとしているので, 所有のコンピュータの処理能力では全画面処理が不可能である. そのため, この式(5)は, 垂直エッジでは, 任意の画面の位置で誤差が生じないこと, 水平線近傍では, 任意のエッジで誤差が生じないことを示している, 補正のほとんど要らない水平線近傍の上下16画素の領域を用いて実験を行った.

当該手法の特徴の1つは, 当該研究で取り扱う実験環境の充たすべき4つの条件を設定したことである. その結果, (1)式(1)~(7)の新しい単純な衝突予測式を導出することができた. (2)水平方向の見かけの動きのみを抽出すればよいので, 時刻間の画像の対応点の探索量が少なくなった. (3)窓問題がなくなり, 従来法で

取り扱われていたコーナーなどの特徴点に限らず輝度が一様でないほとんどの点で有用な速度対を求め衝突予測ができるようになった. 問題は, どのくらい精度良く両眼速度対 (δ_L, δ_R) が求まるかである. これについては, 次節で取り扱う.

2.2 衝突部位と衝突時刻の予測実験

当該手法の有効性は, 図4に示されたステレオビデオカメラ (ビュープラス社製: $\alpha = 0.0038 \text{ rad}$ $a = 12 \text{ cm}$ カメラ焦点距離 = 4mm) を備えた簡単なロボット (東京精機製) を使用して試された.

前回発表までに両眼速度対を現有システムで精度良く測定できることが明らかになったので, ここでは, 衝突時刻と部位が精度良くリアルタイムに測定できることを実験的に検証する.



Fig. 4: A simple robot used at this paper.

当該論文では、歩道・校内・室内など人間が普通に行動している環境でロボットが衝突予測と回避を行い目的地に移動する状態を想定している。アパートから駅までの歩行時間の表示は、 $60\text{m}/\text{min}=1\text{m}/\text{sec}$ の速度で計算されているといわれているので、移動体の速度としては、 $30\text{cm}/\text{sec}\sim 3\text{m}/\text{sec}$ に対処できることが望ましい。また、要求される衝突予測時刻と部位の精度は、高いほど良いのであるが、部位の予測は、数cm程度、時刻としては、人間の反応時間約 0.5sec と考えられる。そこで、 $53.5\pm 0.9\text{cm}/\text{sec}$ (低速)と $146.9\pm 5.4\text{cm}/\text{sec}$ (高速)の2台のラジコンカーを用意し、一定の距離からロボットに向かって走らせて衝突させる実験を行った。その際、衝突予測時刻と部位の時刻 t (frame)毎の計測結果をリアルタイムにファイルに書き込むという方法で記録した。衝突実験は、低速度及び高速度ともに5回ずつ行われた。衝突時刻と部位の予測値は、以下の手順で計算された。(1)2.2節で示した方法で両眼速度対(δ_L, δ_R)を水平線上の画素で求める。(2) $|\delta_L|+|\delta_R|>1$ の画素で、(δ_L, δ_R)を式(1)式(2)に適用して衝突予測部位 $X_T(t)$ と衝突予測時刻 $T(t)$ を求める。(3)左端の画素から走査し、両者とも平均値との差が一定値以内の連続する画素群を統合し、1つの対象物とする。(4)対象物の画素数が一定値以上のものの内最も早く衝突するものの値を求める。このような手順で得られた各frameの予測値例を図5に示す。ただし、実験に使用したコンピュータは、ロボットに乗せられたノートパソコンFMV-BIBLO NX90K/T (Pentium 4:3.2GHz)であり、その処理スピードは、 $6.6\pm 0.44\text{frame}/\text{sec}$ であった。図5では、

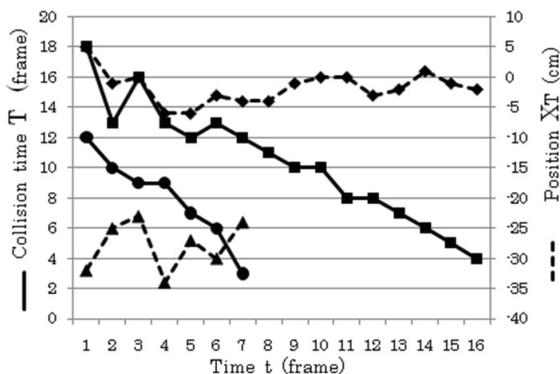


Fig. 5: An example of collision prediction.

衝突予測時刻 T (単位frame)・衝突予測部位 X_T (単位cm)を小数点以下切捨て計算の結果で表示している。衝突時刻の予測については、ラジコンカーの速度がほぼ一定であると思われる範囲(スタート時の数frameを除いた期間)で、衝突予測時刻が線形に変化していること、スタートから衝突までの平均frame数(低速: 20.7frame 、高速: 11.8frame)を正解としたシステムの予測時刻とのずれの平均値と標準偏差は、 $-0.28\pm 1.87\text{frame}$: $-0.042\pm 0.28\text{sec}$ と $1.28\pm 0.78\text{frame}$: $0.19\pm 0.12\text{sec}$ であることが分った。また、衝突予測部位については、測定中の予測部位がほぼ一定であること(図5の点線参照)、衝突部位を衝突予測部位の正解としたとき、衝突予測直前の最後の予測部位と衝突部位の誤差が低速ラジコンカーで $1.8\pm 0.98\text{cm}$ 、高速ラジコンカーで $0.75\pm 1.48\text{cm}$ であった。これらの結果を総合するとほぼ良好な結果であることが分かる。種々の補正やキャリブレーションを行えば誤差はさらに減少するものと思われる。また、衝突直前の数frame処理をしていないのは、両眼視差測定の限界(約 30cm)や視野の限界のためである。

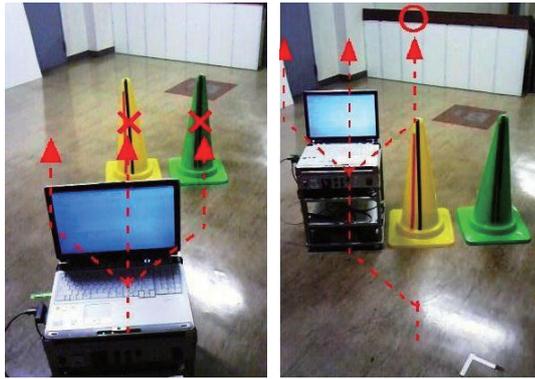
自動車など実用に使用する場合は、安全のため、より詳細な性能検査、カメラの視野の拡大、分解能(1画素の両眼視差に対応する距離 27.4m)の向上、エッジなどの画素単位の対応付け以外の領域間対応の利用、前方に3台の複数のステレオカメラの導入など更なる研究を重ねる必要がある。ほぼ同じ程度の速度で走行している場合や極めて接近して存在する移動体(静止時も含めて)の場合は、両眼速度対は極めて小さくなるので別の距離測定法を併用する必要がある。

3 衝突回避実験例

第2章の実験結果より自己静止の環境では、かなりの精度で衝突時刻と部位をリアルタイムで測定できることが分った。(第2.2節の実験は、等間隔写像における衝突予測式が使用された。)

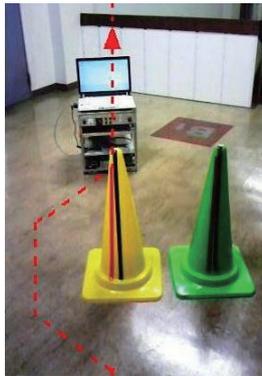
ここでは、衝突予測と回避の実験について述べる。前回は[19]、「移動中のロボットに向かって1台のラジコンカーを走らせた場合、うまく衝突予測と回避ができる」ことを報告した。今回は、複数の衝突対象物が存在する場合の衝突予測と回避行動の実験結果を示す。例えば、複数の対象物があり、このままいくとそのうちの1つにのみ衝突することが予測されたとする。この場合、衝突すると予測された対象物を回避することによって他の対象物に衝突することは許されない。このような課題の1つを図6に示す。図6では、このまま行けば中央のコーンのみ衝突する。右に、例えば当該ロボットの性能上の限界である3秒以内に、回避すれば中央のコーンとの衝突は避けられるが、他のコーンと衝突してしまう。当該システムでは、すべての対象物の行動が抽出されているので、回避する場所に6秒以内に別の対象物が来るかどうかを予測すればよい。図6はそのような場合にうまく対処できることの1つの実験結果である。

次の実験では、中央のコーンあるいは右側のコーンの代わりに人間を後方から接近させる場合(図7参照)が試された。図7では、ロボットの移動目的場所は緑のコーンであり、まず、目的の場所を探索しその位置を求める。その後、衝突を避けながら緑のコーンの手前 50cm で停止することがこのロボットに課せられた



a. Start scene

b. Avoidance after judging of 3 sec before collision



c. Return to the original line

Fig. 6: An example of collision prediction and avoidance.



Fig. 7: Collision avoidance for the human of moving to the left side of the red cone.

仕事である。具体的には、①最初目的地に向かうため目的地の方向を向く、②正面に何もないので直進を始める、③途中で人間が進路に向かってくるが、3秒以内（ロボットの回避限界時間）に衝突することを認知すると、④6秒以内に衝突する空き場所を見つけそこを經由して記録されている目的位置に向きを変え直進する。以上を繰り返して目的を達成する。過程③において、「視野内のすべての対象物に対してそのままロ

ボットが前進したら、前額平行面に何秒後に衝突するか」が計算されている。この情報に従って行動は決められている。この状況判断は常に100msに1回行われるので、もし対象物が、行動を変えればそれに対応して行動計画は変更される。衝突が避けられない場合は、ロボットは停止し、安全になったとき行動は再開される。

その他、左右とも対象物で塞がっている場合も試された（この場合は、衝突回避のため、しばらく静止させた）。その他、目的地の設定として座標値を与える場合と色彩を与え、移動中に静止あるいは移動する障害物を配置する場合も試された。

4 まとめ

最も困難であるといわれている「自己および対象物がともに静止あるいは移動の状態にある衝突予測環境」において、窓問題に頑強な衝突予測をする手法として「両眼速度対法」を提案し、その有効性を確かめるためステレオビデオカメラを備えた2輪の車型移動プラットフォームを構成した。従来研究の内、速度分布を使用する場合、窓問題のため膨大な数の重要な直線的エッジなどを対象外とし、コーナーなどの特徴点に限って動きが測定されてきた。

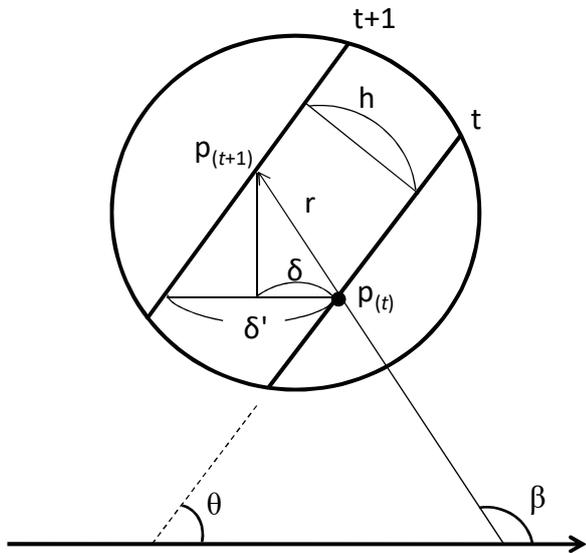
当該研究で新たに行ったことは、(1)窓問題のため捨てられていた重要なエッジを全画面において衝突予測に使用可能な理論(両眼速度対法)を提案したこと、(2)等間隔写像のもと簡単かつ精密な衝突時刻と位置の予測式を求め、衝突予測に基づく真の移動方向を正確に求める式を提案したこと、(3)手法の有効性をステレオカメラを備えた簡単な移動ロボットで確かめたことである。ここでは実験環境を「すべての対象物は平面上を移動すること、すなわち空中を自由に移動することは許されないこと」と制限している。しかしながら、車・人の移動などの重要な日常行動はほとんど含まれていること、また、研究報告の多くの実験場面はこのような環境を取り扱っていることを考慮すれば、それによって取り扱える重要な環境はあまり狭くなっていないと思われる。逆に、従来、窓問題のため捨てられていた重要な膨大な数のエッジが取り扱えるようになったことの意義は大きいのではないと思われる。また、当該手法は、ステレオカメラの設置方向は、自己移動方向に限定する必要は無く、任意の方向に設置しても衝突予測式など全く共通である。

今後の目標は、より性能の高いプラットフォームで、環境の種々の状況を判断しながら、遠隔の目的地まで衝突をしないで移動することを実現することである。

この技術は、いろいろな物事に応用が可能で、車椅子に搭載すれば人にぶつかることなく目的地まで行く補助になるであろう。また、車に搭載すれば、衝突事故などの防止に役立つはずである。このように両眼速度対法は応用範囲の広い技術である。

付録

窓問題：時刻 t の点 $P(t)$ が時刻 $t+1$ の直線上のどの点に移動したか不明。



見かけの水平移動 δ' と真の水平移動 δ の関係は、以下のようになる（真の移動方向角 β は未知）。

$$h = \delta' \sin \theta, \quad h = r \sin(\beta - \theta), \quad \delta = r \cos \theta,$$

の3式より以下の関係式が導出される。

$$\delta = \delta' / (1 - \tan \beta / \tan \theta),$$

($\delta = \delta_L$ or δ_R , $\delta' = \delta'_L$ or δ'_R , $h = h_L$ or h_R , and $r = r_L$ or r_R)

参考文献

- 1) 佐藤淳：コンピュータビジョン，コロナ社，pp.114-115, 1999.
- 2) Nakayama, K. and Looms, J.: Optical velocity patterns, velocity-sensitive neurons, and space perception, Perception, Vol.3, pp63-80, 1974.
- 3) Waxman, A. M. and Duncan, J. H.: Binocular image flows: steps toward stereo-motion fusion, IEEE Trans. PAMI, PAMI-8, pp.715-729, 1986.
- 4) Gross, E. and Tistarelli, M.: Active/dynamic stereo vision, IEEE Trans. PAMI, PAMI-17, pp.868-879, 1995.
- 5) 長田昌次郎：静的及び動的ランダムパターンによる両眼立体視機構の検討，医用電子と生体工学，Vol.17, pp.798-799, 1979.
- 6) 森晃徳：運動物体の視覚情報処理に関する心理学的研究の現状，電総研調査報告書，第207号，pp.31-43, 1982.
- 7) 加賀美聡，岡田慧，稲葉雅幸，井上博允：実時間三次元オブテクアルフロー生成システム，第17回日本ロボット学会学術講演会，pp.29-30, 1999.
- 8) 依田育士，細谷大輔，坂上勝彦：ユビキタスステレオビジョンによる滞留人計測システム，映像情報メディア学会誌，Vol.60, No.11, pp.1829-1834, 2006.
- 9) 清水早苗，山本和彦，王彩華，佐藤雄隆，棚橋英樹，丹羽義典：全方位ステレオシステムを搭載した移動体による実環境からの動物体検出，電気学会論文誌C，Vol.124, pp.1288-1295, 2004.
- 10) Regan, D. and Beverley, K. I.: Binocular and monocular

stimuli for motion in depth: changing-disparity and changing-size feed the same motion-in-depth stage, Vision Res, Vol.19, pp.1331-1342, 1979.

- 11) Richards, W.: Structure from stereo and motion, J. Opt. Soc. Amer., Vol.A2, pp.343-349, 1985.
- 12) Prazdny, K.: Ego-motion and relative depth map from optical flow. Biological Cybernetics, Vol.36, pp.87-102, 1980.
- 13) Kobayashi, H. and Sugie, N.: Decomposition of spherically projected optical flow into translational and rotational components, Nagoya University Technical Report, No.8403, 1984.
- 14) Mori, T.: An active method of extracting ego-motion parameters from optical flow, Biological Cybernetics, Vol.52, pp.405-407, 1985.
- 15) Regan, D. and Cynader M.: Neurons in cat visual cortex tuned to the direction of motion in depth: effect of stimulus speed, Investigative Ophthalmology & Visual Science, 22, pp.535-550, 1982..
- 16) Toyama, K. and Kozasa, T.: Responses of Clare-bishop neurons to three dimensional movement of a light stimulus, Vision Res., Vol.22, pp.571-574, 1982.
- 17) 森文彦，山田博三：動的衝突予測法，公開特許公報，2005-182121, 2005.
- 18) Mizuno, M., Yamada, H., Shibata, N., Uno, K., Taguchi, K., Mori, F. and Mori, T.: A method of collision prediction by binocular velocity pair under the condition of ego and object motion, Proc. of IEVC2007, 2007.
- 19) 森文彦，水野真，山田博三：両眼速度対による自己と対象移動環境での衝突予測，DIA2006 講演論文集，pp.102-105, 2006.

大豆の子実検知を行う画像センシング手法の開発

○山口幹太 小澤誠一 北園淳 (神戸大学) 吉田武史 (青山学院大学) 大川剛直 (神戸大学)
村上則幸 辻博之 (北海道農業研究センター)

Development of an Image Sensing Method to Detect Grains of Soybeans

*K.Yamaguchi, S.Ozawa, J.Kitazono(Kobe University)

T.Yoshida(Aoyama Gakuin University), T.Ohkawa(Kobe University)

N.Murakami and H.Tsuji (Hokkaido Agricultural Research Center)

Abstract— Recently, “Smart Agriculture” that is agriculture with Information and Communication Technology (ICT) is receiving a lot of attention. “Smart Agriculture” uses computers to control crops’ growing environment and status, and aim to streamlining agriculture and increasing yield. It is thought that grains are important growing information directly connected with yield. In this paper, we handle soybeans as crops, line up detecting candidate regions by grains’ shade information from one picture, and detect soybeans using Convolutional Neural Network (CNN) based on RGB pixels information. In the experiments, using the pictures taken at the actual farm, the proposed system is evaluated based on the detection accuracy of soybeans. The F-measure of soybean detection is more than 80% for the training dataset and the testing dataset.

Key Words: Smart Agriculture, Image Processing, Detecting Grains

1 はじめに

近年、農業に ICT 技術を用いる「スマートアグリ」が注目されている¹⁾。ICT 技術を用いることで、これまで長年培った技術と勘を頼りに行っていた農業から、蓄積した生育データや気象データを用いて解析し、そこから得られた知見を元に農作業の効率化を図り、また農場の水分や肥料などの管理、収穫といった重労働をロボットに行わせて省力化することが期待されている。例としては農業作業や無線 LAN 搭載の農業機械から情報を収集し、圃場一区画ごとにデータを記録し、その情報を元に最適な肥料などの量を提案するシステム²⁾が提供されている。

スマートアグリでは生育情報収集の方法の一つとして、低労力で多くの情報を得ることができる画像センシングが活発に利用されている。先行研究では、ドローンを用いて稲の圃場を上空から複数撮影し、画像から三角測量の原理を用いて 3 次元モデルを生成して稲の高さを調べる手法⁴⁾がある。また子実の検知の先行研究としては、村上ら⁵⁾は木になったままのナシの果実の画像を撮影し、果実の陰影情報と輪郭情報より計数する手法や、X.Liming ら³⁾はイチゴの色から子実領域を抽出し、エッジ情報を元にして計数と形状把握する手法を提案している。このスマートアグリの取り組みの一つとして、本論文では大豆の収量推定を行うシステムを開発することを最終目的とし、画像処理技術を活用して大豆の生育因子を自動抽出する研究を行う。

本研究では、環境に伴う成長度合において不明な点の多い農作物である大豆を扱い、生育情報としては収量に直結すると考えられる子実の検知を行う。そのために、画像処理の分野において広く利用されている基礎技術を組み合わせることで生育情報の抽出方法を開発する。まず子実の陰影に着目し、Haar-Like 特徴という輝度に注目した局所特徴量を用いて、AdaBoost で学習を行っ

た識別器に子実の検知漏れのないように候補領域を挙げさせる。次に挙げられた候補領域から子実以外の領域を棄却するために、輝度値や色情報を正規化した画像を Deep Learning の一種である Convolutional Neural Network (CNN)⁹⁾ に学習させて、識別させる手法を提案する。

本論文では以下の構成で述べる。2 章では提案手法と用いられた技術、実験方法と手順について述べ、3 章では実験結果と考察、4 章で結論を述べる。

2 提案手法

梨や苺などの色もしくは影の形が茎や他の部位と大きく異なっている農作物の子実に対して適応した手法が確立されているが、今回扱う大豆の子実は茎や葉と同じ色味を帯びており、また影の形も茎や葉の凹凸に非常に類似しているため、色情報もしくは影情報のみを用いた一つの識別器で検知することが困難である。

そこで提案手法では Course-to-Fine のアプローチに基づいて、Fig.1 のように 3 ステージの処理を行い検知する。はじめに陰影情報に基づいて検知する候補を挙げる「探索ステージ」、次に輝度やコントラストなどの撮影環境に対してロバストにするための「正規化ステージ」、最後に正規化された候補領域を色情報に基づいて識別する「精査ステージ」がある。

2.1 探索ステージ

色情報だけで背景を含めた農作物の画像から子実を検知することは、他の部位を誤検知してしまい困難なため、子実の作る影に注目して陰影情報から候補領域を挙げる。

本研究では、子実の検知に画像検知手法の一つである、「Viola-Jones の手法」⁶⁾を用いる。Viola-Jones の手法は、もともと顔の陰影に注目した検出手法として提案されたが、現在は広く物体検知に応用されている。Haar-Like 特徴を用いて特徴量を抽出し、AdaBoost で

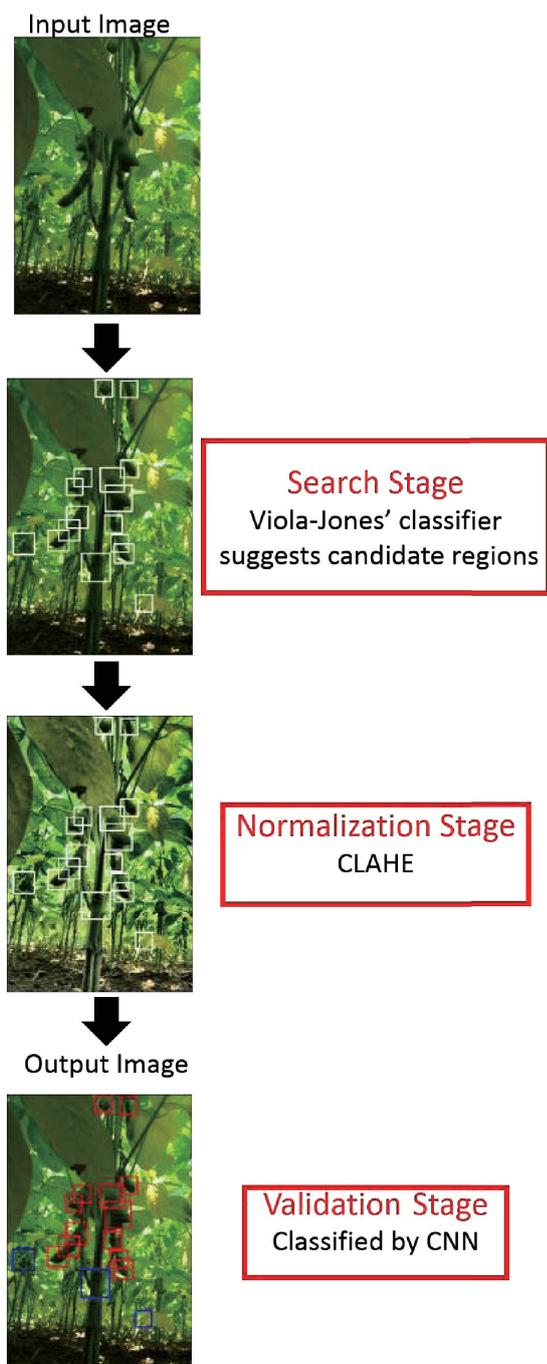


Fig. 1 Over view
 White rectangles mean candidate regions suggested by Viola-Jones' classifier.
 Red rectangles mean regions classified to be a grain by CNN.
 Blue rectangles mean regions classified not to be a grain by CNN.

学習を行い、カスケード構造の識別器を構成することで画像データから高速に対象物体を抽出する手法である。

この手法を子実に適応するため、事前に用意された Fig.2 のような特徴を正規化を行っていない子実や茎や背景などの学習画像に対して各々当てはめる。そして Fig.3 のように黒色矩形内と白色矩形内の平均輝度値の差のみで判定する弱識別器として Haar-Like 特徴を用

いる。学習データを判定させた後にエラー率に基づいて重みを与え、弱識別器の重み付き多数決をとることで最終的な識別結果とする強識別器を構成する。

検知対象画像に対して移動窓⁸⁾を用いてスケールを変えながら先程の強識別器に識別させて、候補領域として「精査ステージ」に受け渡す。

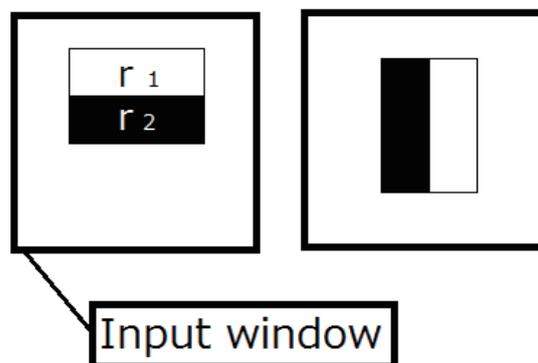


Fig. 2 Examples of Haar-Like feature

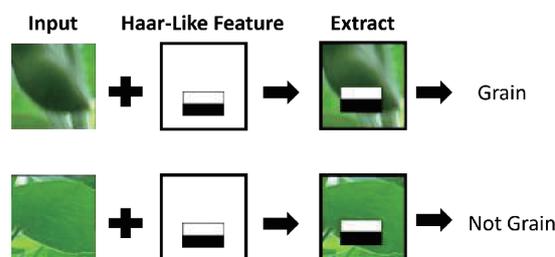


Fig. 3 A example of extracted features

2.2 正規化ステージ

天候の変化や透光度合いによって、検知対象画像によっては領域ごとに異なる輝度や色合いになってしまう。「探索ステージ」で用いられている特徴は周辺輝度値との相対性が導入されているために、撮影環境に対してロバストであるが、次の「精査ステージ」では輝度と色情報の正規化が必要となる。そのような場合に有効であるのが局所領域ごとに正規化を行う局所適応的ヒストグラム均一化 (Adaptive Histogram Equalization; AHE) である。中でも局所領域内で輝度値が極端に偏っていたとしても対応可能な CLAHE (Contrast Limited AHE) を導入する。Fig.4 の左図のように上部と下部とで輝度と色合いが異なる場合に対して、CLAHE を行うことで右図のように正規化された画像を得ることができる。



Fig. 4 CLAHE

2.3 精査ステージ

探索ステージでは子実と茎とを誤検知する例が非常に多くあったため、色情報を導入することで主に茎部分を棄却する。また CLAHE を行うことで土部分と子実の色味が似てしまうために、探索ステージでは予め棄却し、検知対象外にすることも目的としていた。

CLAHE を用いて正規化を行った入力画像内の探索ステージで挙げた候補領域に対して、子実の色情報を学習させた CNN¹⁰⁾ で精査する。CNN の学習には膨大な量の学習データが必要であるが、本研究では得ることの出来る大豆の子実の枚数が必要数に満たないため、事前学習したモデルに対して転移学習を行うこととした。「ImageNet2012」の訓練データで学習済みの CaffeNet のモデル「bvlc_reference_caffenet.caffemodel」が、1000 クラスの分類問題に対して accuracy が 57.4% という高精度であったため、CLAHE で処理した子実画像を用いて転移学習させることとした。CNN の構造は Fig.5 に示す。

2.4 評価方法

子実の検知には苗に接近して横から撮影した画像（群落画像）を用いる。事前に群落画像に子実部分を正例、背景などを負例としてラベル付けを行い、トレーニングデータの中で子実の全貌が写っている一部の画像を学習に、テストデータは全てを評価対象とする。

評価方法は群落画像を入力し、Fig.6 のように識別器が検知した領域の中心と正例の小領域の中心との距離 L 、検知領域の縦幅 RH 、横幅 RW 、ラベルの縦幅 LH 、横幅 LW として、 $(L < \frac{1}{2}\sqrt{LW^2 + LH^2})$ かつ $(\frac{1}{2}\sqrt{LW^2 + LH^2} < \sqrt{RW^2 + RH^2} < 2\sqrt{LW^2 + LH^2})$ を満たすものを検知できたとし、そうでなければ検知できなかったとする。

3 実験

3.1 実験環境

本実験に用いた画像データは、北海道芽室市の大豆畑において 2014 年 6 月から 9 月までのデータを使用し、2 章で述べた手法を用いて生育情報を抽出を行った。

画像データの撮影には Fig.7 の短軸ロボットに Fig.8 のようにロボットに仰角 $\theta = \tan^{-1}\frac{1}{3}$ [rad] つけてカメラを取り付け、下から上へと移動させながら撮影を行った。

3.2 実験方法

北海道の芽室市において、2014 年 7 月から 8 月にかけて、2 ブロックで撮影された連続画像について実験を行った。撮影画像の解像度は 2592×1944 であり、427 枚の群落画像から正例としてラベル付けした 11000 枚の内の 7286 枚の子実画像と、子実以外が写っている 7165 枚の画像を負例としてトレーニングデータとした。同様に 75 枚の俯瞰画像から 2326 枚の正例と 1379 枚の負例をラベル付けし、テストデータとした。Viola-Jones の手法を用いた識別器には原画像を、CNN には CLAHE を施した画像を与えた。

子実を正しく検知できた場合を Hit、検出できず見逃してしまった場合を Miss、子実以外を検出してしまっ

た場合を False Alarm (FA) と呼ぶ。検出を行う際に子実の大きさが 70×70 から 200×200 ピクセルに収まっているという先見知識を用いて、検出時にはその制限をかけている。探索ステージで子実の見落としのないように Viola-Jones の手法を用いた識別器に対して、各カスケードでの正例と識別器が判別したサンプル中の負例の割合を許容する最大値 $f = 0.5$ 、全正例数の中で正例と判別できた割合を許容する最小値 $d = 0.99$ と設定して見落としを減らすようにし、画像サイズは 24×24 とした。

3.3 実験結果と評価

提案手法によって群落画像から子実を検出し、生育情報として抽出できていることを 2.4 項の指標をもって評価する。

まず撮影環境を正規化する CLAHE の有効性を示す。Table.1,2 から CNN で子実を判定する際に CLAHE を前処理として用いることで、Recall, Precision 共に上昇しており、F-measure がトレーニングデータとテストデータで平均して約 7% 向上した。この結果から、撮影環境による色や輝度の違いを CLAHE を用いて正規化することが有効であることが分かった。

次に Viola-Jones の手法と提案手法を比較することで、提案手法の有用性を示す。Table.3,4 に示すとおり、Viola-Jones の手法だけでは Recall に比べて Precision が低く、誤検知が多いため F-measure も 80% を下回っている。提案手法では CNN を Precision を上げるために導入しているため、Precision がトレーニングデータ、テストデータ共に 10% 以上向上しており、陰影情報だけでなく色情報も検知において重要な情報であることが分かった。本実験では、生育情報の抽出には誤検知を減少させることよりも、見落としを減少させることを重視するが、Precision の向上に対して、Recall の下降度が比較的低いため、提案手法は有用であると考えられる。

テストデータに比べてトレーニングデータの F-measure の大きな向上が見られるが、これは Viola-Jones の手法でテストデータの Precision がトレーニングデータに対して元々高かったことに起因すると考えられる。

テストデータとトレーニングデータでの Precision と Recall の差を解析するために実験に用いた画像を調べたところ、Fig.9 のような低輝度背景の画像がトレーニングデータに比較的多く、Fig.10 のような高輝度背景の画像が多くあったため、背景の輝度値による結果の違いを Table.5,6 に示す。群落画像には、トレーニングデータとテストデータそれぞれ高輝度背景画像が 129 枚と 43 枚、低輝度背景画像が 298 枚と 32 枚であった。高輝度背景では子実を含め全体的に輝度が高く影が出来にくいために Viola-Jones の手法での見落としが増えており、Recall が大きく下がってしまっている。

また、Fig.11 のように CNN で棄却しているものの誤検知の大半は茎である。色と影のつき方が子実に非常に似ているため、棄却できなかった候補領域たちの CNN の出力を解析すると子実に近い値が大半であった。

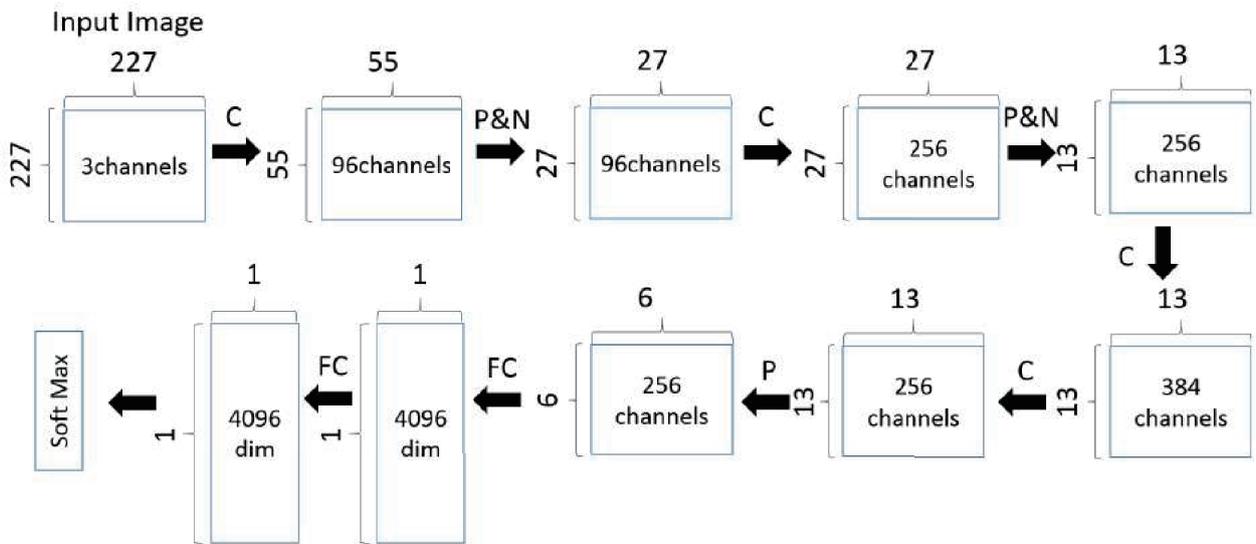


Fig. 5 Structure of CNN(C:Convolution P:Max Pooling N:Normalization FC:Fully Connect)

Table 1 Accuracy of CNN with CLAHE

	TP	TN	FP	FN	Precision	Recall	F-measure
Training	7146	6460	705	140	91.0%	98.1%	94.2%
Test	2159	1215	164	167	92.8%	92.8%	92.8%

Table 2 Accuracy of CNN without CLAHE

	TP	TN	FP	FN	Precision	Recall	F-measure
Training	6610	5607	1558	676	80.9%	90.7%	85.5%
Test	2009	1701	299	319	87.0%	86.4%	86.7%

Table 3 Result of Detecting Grains by Viola-Jones' Classifier

	Hit	Miss	FA	Precision	Recall	F-measure
Training	10279	721	4498	70.0%	93.4%	79.8%
Test	1885	441	533	78.0%	81.0%	79.5%

Table 4 Result of Detecting Grains by Proposal Method

	Hit	Miss	FA	Precision	Recall	F-measure
Training	10012	988	1650	85.9%	91.0%	88.4%
Test	1798	528	167	91.5%	77.3%	83.8%

Table 5 Result of Detecting Grains in high brightness background

	Hit	Miss	FA	Precision	Recall	F-measure
Training	2003	527	407	83.1%	79.3%	81.1%
Test	836	329	79	91.4%	71.8%	80.4%

Table 6 Result of Detecting Grains in low brightness background

	Hit	Miss	FA	Precision	Recall	F-measure
Training	8009	461	1243	86.6%	94.6%	90.4%
Test	962	199	88	91.6%	82.9%	87.1%

4 結論

本論文では、画像処理により農作物の生育情報を抽出し、収量予測を行うシステム開発を最終目標とし、大豆の子実の検知により生育情報の抽出を行った。実験では、Haar-Like特徴、AdaBoost学習、カスケード構造を組み合わせる Viola-Jones の手法を用いた陰影情報だけ

による検知では誤検知が多いため、CNN を組み合わせ色情報を用いることで子実の検知を行った。また、撮影環境による輝度や色の変化を正規化するために、CNN に識別させる画像には CLAHE を用いて前処理を行うことで精度が向上した。

実験結果より、本論文で提案した手法は子実を検知することができ、F-measure が 80% を超える結果となっ

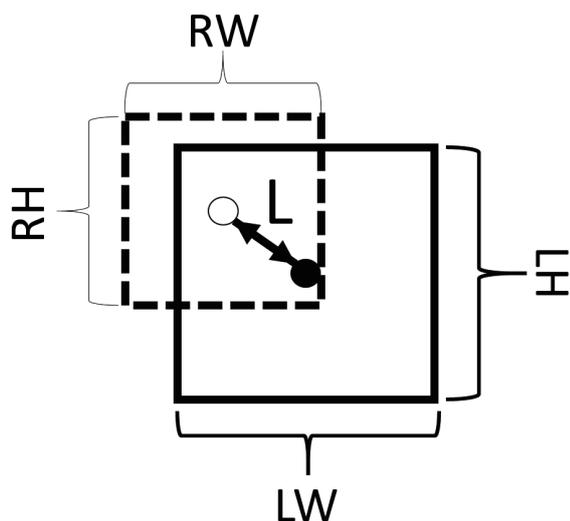


Fig. 6 Evaluation method



Fig. 7 Single-axis robot



Fig. 8 Method for photographing images



Fig. 9 Result Image of low brightness background
 Yellow rectangles mean regions labeled positive example
 Red rectangles mean regions classified to be a grain by CNN
 Blue rectangles mean regions classified not to be a grain by CNN



Fig. 10 Result Image of high brightness background
 Yellow rectangles mean regions labeled positive example
 Red rectangles mean regions classified to be a grain by CNN
 Blue rectangles mean regions classified not to be a grain by CNN



Fig. 11 Result Image containing many False Alarms
 Yellow rectangles mean regions labeled positive example
 Red rectangles mean regions classified to be a grain by CNN
 Blue rectangles mean regions classified not to be a grain by CNN

た。しかし茎の誤検知がいくらかあり、学習データに負例として茎の画像枚数を増加させるなどで改善が見られないか試していく必要がある。また低輝度背景の画像に比べ、高輝度背景の画像に対する見落としが多いため、検知手法に写真の輝度に依存しない方法を検討していく。

また本論文では子実が茎や葉に隠れて全て写っていないことや、日向や日陰が混在している画像が実験結果に影響を及ぼすことがあったため、今後そういった問題が起こらないような撮影方法を検討していく必要がある。

5 謝辞

本研究は、平成 27 年度農林水産省委託プロジェクト研究「多収阻害要因の診断法及び対策技術の開発」の助成を得て行われた。

参考文献

- 1) NEC Corporation:農業 ICT ソリューション: ソリューション・サービス — NEC,NEC(Japan), <http://jpn.nec.com/solution/agri/>
- 2) kubota Corporation:KUBOTA Smart Agri System, NEC(Japan), <https://ksas.kubota.co.jp/>
- 3) X.Liming, and Z.Yanchao:Automated strawberry grading system based on image processing,Computers and Electronics in Agriculture 71,32/39(2010)
- 4) 土佐光司:北陸スマートアグリプロジェクト, http://www.kanazawa-it.ac.jp/prj/coc/activity/h27/prj_smart_agri.html
- 5) 村上則幸, 吉田睦, 吉田武史, 塩澤秀門, 深尾隆則:ナシ計数のための画像処理手法の研究, システム制御情報学会研究発表講演会講演論文集, 57/62 (2013)
- 6) P.Viola and M.Jones:Robust Real-Time Face Detection,International Journal of Computer Vision 57(2), 137/154 (2004)
- 7) K.Zuiderveld:Contrast Limited Adaptive Histogram Equalization, Graphic Gems IV. San Diego: Academic Press Professional, 474/485(1994)
- 8) R.Szeliski:Computer Vision Algorithms and Applications, 共立出版株式会社, (2013)
- 9) C.Bishop:Pattern Recognition and Machine Learning, Springer, 1/267(2010)
- 10) 中山英樹:深層畳み込みニューラルネットワークによる画像特徴抽出と転移学習, 電子情報通信学会音声研究会 7 月研究会, (2015)

大豆の花検知と花数計測を行う画像センシング手法の開発

○大西哲 北園淳 小澤誠一 (神戸大学)
吉田武史 (青山学院大学) 大川剛直 (神戸大学)
村上則幸 辻博之 (北海道農業研究センター)

Development of an Image Sensing Method to Detect and Count Flowers of Soybeans

*T. Onishi, J. Kitazono and S. Ozawa (Kobe University)
T. Yoshida (Aoyama Gakuin University), T. Ohkawa (Kobe University)
N. Murakami and H. Tsuji (Hokkaido Agricultural Research Center)

Abstract— In this paper, we propose an image sensing method to detect flowers and count the number of flowers in the crowd of soybean plants for a smart cyber-physical system in agriculture. To count all flowers on a soybean plant, multiple pictures are taken by a digital camera attached to a single-axis robot, which is moving from bottom to top of a targeted plant. In the experiments, we observed soybean plants at 18 observation points in the field of Memuro town, Hokkaido, Japan from July 16th to August 3rd, 2015, and collected 1207 images which contain at least one flower in total. The experimental results demonstrate that the propose system has fairly good accuracy in both flower detection and flower count. Both recall and precision rates of flower count are over 0.80.

Key Words: Flower counting, Image sensing, smart agriculture

1 はじめに

近年、日本では農業従事者の高齢化と後継者不足が大きな問題となっている。これは、ノウハウがなければ新たに農業に取り組んだ際、早期から収益を上げることが難しく、また農作業自体に非常に労力がかかることが理由の一つとなっている。そのため、経験がなくとも農業において十分な利益を得るために、熟練農家の持つノウハウを若い世代へと継承することは重要である。近年、農業に ICT (Information and Communication Technology) 技術を取り入れたスマート農業が注目を集めている。このスマート農業とは、農作物の生育環境として気温、湿度、日照量、土壌水分量等をセンサーで取得し、農作物にとって最適な環境となるようヒーターや窓、水分や肥料をコンピュータで管理して栽培し、省力で収量の増加につなげることで農作業の効率化を図るものである。スマート農業は、最終的な収量と農作物の環境のデータの蓄積を行うことでノウハウのデータ化を進めている。これは農業の省力化だけでなくノウハウの継承の手助けとなっている¹⁾²⁾。

スマート農業をより効果的に行うためには、生育環境と農作物の成長を表す生育情報の二つのデータの蓄積が不可欠である。現在、実際に提供されているサービスにおいて、気候等の生育環境をセンサーで取得することができるが、花数や分枝数等の生育情報は画像の目視確認や実際に引き抜いて人手による計測が中心となっている。そのため、生育情報の取得を大規模に行うことが難しく、生育情報の蓄積が困難となっている。そこで、人手に頼らず機械が自動的に生育情報の観測を行うことが求められている。生育情報の中でも、花が受粉することにより子実ができることから、花数を求めることは収量の予測に役立つと言える。

花領域の取得を行う既存の研究には、Najjar と Zagrouba の手法³⁾がある。画像を Lab 色空間で表し、L, a, b それぞれのヒストグラムに対して大津の二値化処

理を行い、花領域の分離を行う。その際、二値化による結果を L, a, b の 3 つ得る。この中で最も上手く花領域と背景を分離した画像を選択する必要がある。また、花数計測の先行研究である Biradar らの手法⁴⁾では、俯瞰画像を利用し、ローパスフィルターと二値化、エッジ検出を用いて高い識別率を出している。しかし俯瞰画像を使用するため、花が上部に咲いている場合のみ対応し、主茎に近い位置に花ができる作物には対応できない。また俯瞰画像では、一株を特定できないことから、一株に限定した生育情報の取得は難しい。

農作物の生育情報を観測する上で、群落内における生育情報の取得は、特定の株に限定することで、株の成長過程が得られるため有用である。また花の位置が側面や主茎に近い植物に利用が可能である。これまで群落内での撮影では、農作物とカメラの間に十分な距離が取れず、農作物の全体を一枚の画像で撮影することができないため困難であった。また群落内では接近して撮影を行うため、葉による遮蔽が発生することが多く、一度の撮影ではすべての情報を得ることができない。そこで上下に稼働する短軸ロボットを用いて、農作物を下から上へと複数枚の画像を撮影し、複数枚の画像の重なりを利用することで、農作物の全体を撮影し、複数画像の中で遮蔽の起こっていない画像を得る。

本論文では、農作物の生育情報を自動的に取得し、収量の予測を行うシステムを最終目標として、群落内における生育情報の取得のための画像処理技術の開発を行う。本研究では農作物として大豆を扱い、生育情報として有用な花検知を行い、一株当たりの花数を求める。一画像からの花検知では、領域分割と色相に基づく判定により、花領域の取得を行う。一画像からの花検知で得られた花領域から、特徴点の抽出を行い、特徴量を計算し、農作物の下から上へと撮影した連続画像における、画像間の上下関係を用いて対応付けを行う。花領域の対応付けを行うことで、一株当たりにお

ける花数を計測する。実験は、北海道芽室市の大豆畑において、群落内に入り単軸ロボットを用いて農作物を下から上へと撮影した連続画像 100 系列 1207 枚の画像に対して行い、検知した花と目視で確認した花との比較を行う。

2 一画像から花検知を行う画像センシング手法

一画像からの花検知を行う画像センシング手法について述べる。花検知は、画像を領域分割により小領域へと分割する。この小領域それぞれに対して色相に基づく判定を行い、花領域の取得を行う。隣接する花領域の統合を行い、過分割された花領域を一つの領域へと統合し、再度色相に基づく判定を行う。最後に識別器を用いて判別を行い、花領域を得る。

2.1 SLIC による領域分割

花には複数の色が含まれており、色相に基づく判定で対象となる色のみを指定して画像から抜き出すだけでは、花領域の一部のみを抜き出す形となり、一つの花から複数の領域が得られ正確な花数の取得や、花領域全体を得ることが難しい。そのため、まず領域分割を行い花と茎や葉の分離を行い、小領域単位で処理を行うことで、花領域内部での色の変化の影響を受けずに花領域全体を得る。

画像の領域分割手法として本研究では Simple Linear Iterative Clustering (SLIC)⁵⁾ を用いて画像に対して領域分割を行う。SLIC は色の類似度と画素の位置関係に基づいてクラスタリングを行うため、花と周囲の色の違いを利用する本手法において有用である。SLIC は、はじめに画像をグリッド状に区切り、それぞれのグリッドを一つのクラスタとし、これら各クラスタの中心 $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ ($k = 1, \dots, K$) を計算する。このとき (l_k, a_k, b_k) は CIELAB 色空間における座標、 (x_k, y_k) は画像内の座標とする。この中心 C_k から、初期グリッドの大きさ S の 2 倍の範囲のピクセル $2S \times 2S$ について、次式を用いて色と画像内の座標を用いた距離関数 D を求める。

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (1)$$

但し、

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (2)$$

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (3)$$

式(1)-(3)の、は色と位置の重みである。この距離関数が最小となるクラスタにピクセルを割り当て、再度クラスタの中心を計算し、ピクセルの割り当てを行う。この割り当てが変化しなくなるまで計算を繰り返し行うことで領域分割を行う。

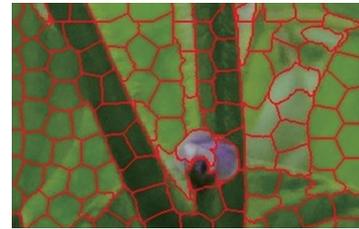
実際に SLIC を適用した画像を Fig. 1 に示す。領域分割により、葉や茎と、花の領域が分離していることが確認できる。

2.2 色相に基づく花領域の判定

大豆の花に紫色が多く含まれているため、紫色であるピクセルを求め、各小領域内での紫ピクセル数によ



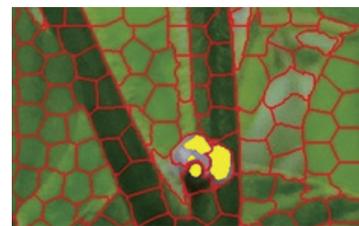
(a)



(b)

Fig. 1: An example of SLIC result: (a) a flower image and (b) an image segmentation result.

り花領域であるかの判別を行う。RGB 色空間では紫色の指定が困難であるため、より直観的に色を指定することのできる HSV 色空間に変換し、紫色となる色相 H の範囲を定め、定めた範囲内に色相の値が収まるピクセルについて紫ピクセルとする。領域分割で得られた小領域について、紫ピクセルの数を計測する。この紫ピクセルの数が閾値以上の場合、その小領域を花領域とする。Fig. 2 に例を示す。



(a)



(b)

Fig. 2: An example of Flower classification based on purple pixels: (a) an image whose target purple pixels are colored by yellow and (b) an image of flower counting result.

2.3 隣接領域の統合処理

領域分割で、一つの花が複数の小領域に分割される場合がある。このとき、それぞれの小領域が花領域と判定された場合、実際の花数より多く検知してしまう。そのため花領域が隣接している際、その領域を一つに統合する必要がある。領域が隣接しているかは、隣接するピクセルに SLIC の異なる小領域に属するピクセ

ルが存在するかにより知ることができる。この処理を花領域について行い、花領域同士が隣接していた場合その二つの花領域を統合する。この処理を繰り返し、統合した領域内の紫ピクセルの数を再度計測し、閾値以上であれば花領域であるとして一画像から花領域の個数を求める。

Fig. 3 では、統合処理によって花の数が正しく検知されていることがわかる。

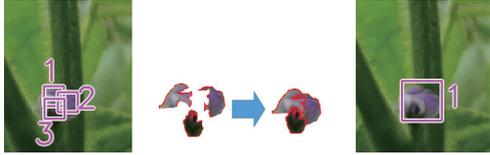


Fig. 3: Three detected flower regions for one flower are merged into one flower region.

2.4 Convolutional Neural Network を利用した識別器による判定

花領域を色情報のみで取り出すと、対象とする株以外の花や、パープルフリンジ等による誤検知が見られる。対象とする株の花とそれ以外の誤検知の判別を行うため、識別器による判定を行う。対象とする株の花は、花の成長段階、枝の張り出し、カメラとの角度、咲く向きによって、見え方が大きく異なる。これらの見え方の異なる花画像を Convolutional Neural Network (CNN)⁶⁾ を用いて学習させ、識別を行う。本研究では、CNN を畳込み層 5 層、プーリング層 3 層、正規化層 2 層、全結合層 3 層から構成する。活性化関数は Relu¹⁰⁾ を使用する。画像から花と、花以外の葉、茎、距離の遠い株等を抜き出し、花と花以外の 2 クラス分類を行う。

3 一画像からの花検知に基づく一株当たりの花数計測を行う画像センシングシステム

一画像からの花検知に基づく一株当たりの花数計測を行う画像センシングシステムについて述べる。連続画像の撮影を行い、一画像からの花検知で得られた花領域から、特徴点の抽出を行い、特徴量を計算する。これらの特徴量に対して、カメラの撮影間隔と短軸ロボットの速度、対象の株までの距離を用いて、連続画像における花の写る位置の移動先を計算し、予想される移動先に範囲を絞り、対応付けを行う。花領域の対応付けを行うことで、連続画像内において複数回同じ花が撮影される場合に、対応を持つ花を一度のみ計測し、重複して計測せず、正確な花数を計測する。

3.1 農作物の連続画像撮影方法

撮影では、約 1m の単軸ロボットを使用した。単軸ロボットは、台を任意の速度で上下が可能であり、台にカメラを装着し撮影の時間間隔を設定することで、農作物を下から上へと一定間隔での撮影を行う。

今回は、台の速度を毎秒 1cm、撮影の時間間隔を 3 秒ごとに、一回の撮影で 30 枚の画像を撮影し、高さ 3cm ごとの農作物の画像を得ている。また、カメラは横向きに装着している。画像同士のオーバーラップは 85% となった。このとき画像内に複数の株が写るため、対象となる株の周囲のみを検知範囲とし、連続画像における重なりを抑えるため、Fig. 4 に示すように検知に利用する範囲を限定した。



Fig. 4: A target region for flower detection.

3.2 花周囲からの特徴点抽出

花検知で得られた花領域について、その周囲から Feature from Accelerated Segment Test (FAST)⁷⁾ を用いて特徴点の抽出を行う。FAST とは、注目画素を中心とし周囲 16 画素の輝度が注目画素よりも高い、もしくは低い画素が連続する場合、注目画素をコーナーとして検出を行う。FAST はテクスチャが複雑となる自然画像に対して、特徴点が多く抽出されることから、特徴点の抽出を花の周囲のみと限定し、限定した狭い範囲内で特徴点を得る必要のある本システムにおいて有用である。Fig. 5 に得られる特徴点を示す。これらの特徴点について、Oriented FAST and Rotated BRIEF (ORB) 特徴量⁸⁾ を計算する。ORB 特徴量は計算が高速であり、複数の株について撮影を行い、一株を連続画像で撮影し、繰り返し特徴量の計算を行う本システムにおいて有用である。



Fig. 5: An image of flower counting: (a) before keypoint detection, (b) after keypoint detection.

3.3 花領域の対応付け

連続画像における、画像間の上下関係から特徴点の対応付けを行う。連続画像において、ある一画像で撮影された花が、次の画像において撮影される位置を、カメラの撮影間隔と短軸ロボットの速度、対象の株までの距離を用いて、連続画像における花の写る位置の移動先を計算する。予想される移動先に範囲を絞り、特徴点の対応付けを行う。特徴点の対応付けの結果、最も多く対応点をもった花領域から対応を決定する。これをすべての連続画像に対して繰り返し、一株あたりの花数の計測を行う。花領域の対応付けにより、一つの花が複数回撮影された場合において、複数回計測されることなく、一つの花として計測が可能となる。この処理を、Fig. 4 に示す範囲において、連続画像内に花が最大 3 度写ることから、二枚先の画像まで対応付けを行う。Fig. 6 に対応付けの結果を示す。

4 一株あたりの花数計測実験

4.1 実験環境

本実験は北海道の芽室市において、2015 年 7 月から 8 月にかけて、3,4 日ごとに 18 ブロックで 1 度ずつ撮影を行った。撮影を行った 18 にブロックはそれぞれ、1-1, 1-2, 1-3, 2-1, 2-2, 2-3, 3-1, 3-2, 3-3, 7-1, 7-2,



(a)



(b)

Fig. 6: An image of flower matching result: (a) an image of matching the one with next image (b) an image of matching the one with after the next image The light blue circle and yellow line indicate detected flowers and matching results, respectively.



Fig. 7: The example image of flower detection in an image.

7-3, 8-1, 8-2, 8-3, 9-1, 9-2, 9-3 とした。このうち 1-1 から 3-3 までのブロックは通常的环境下で生育されているが、7-1 から 9-3 のブロックでは意図的に除草剤を多く与えることで薬害による生育障害を起こした。使用したカメラは SONY Cyber-shot DSC-HX60V である。撮影された画像の解像度は 2592×1944 である。本実験は撮影した画像のうち、目視で花が確認できた期間 7 月 16 日から 8 月 3 日の、農作物の写っている連続画像 100 系列、1207 枚に対して行った。このとき、画像全体の色が大きく異なるものは、事前の処理で棄却が可能であるとして、使用していない。本実験では、2.4 における CNN の学習時には、北海道羊ヶ丘市の農場で得られた、花が目視で確認できた 286 枚の画像から切り出した画像を用いた。本実験において CNN の学習は、利用することのできる花画像が少ないため、一般物体認識の学習済みモデル⁹⁾ をパラメータの初期値として、Fine-tuning を行う。Fine-tuning という言葉は、深層学習において通常、教師なし学習を用いてネットワークの初期値を設定したのちに、学習を行うことを指すが、ここでは大規模データセットをネットワークの初期化に利用することを指す。

4.2 実験結果と考察

撮影された画像のうち、実際に花を目視できた連続画像 100 系列について一株当たりの花数計測を行った。実際の花検知結果画像を Fig. 7 に示す。一株当たりの

Table 1: Results of flower count of a soybean plant

BN	TP	FP	FN	Precision	Recall
1-1	71	7	21	0.910	0.772
1-2	46	0	10	1.000	0.821
1-3	38	1	14	0.974	0.731
2-1	39	9	9	0.813	0.813
2-2	53	2	17	0.964	0.757
2-3	39	2	20	0.951	0.661
3-1	56	9	11	0.862	0.836
3-2	53	4	15	0.930	0.779
3-3	63	4	8	0.940	0.887
7-1	38	13	9	0.745	0.809
7-2	53	2	8	0.964	0.869
7-3	30	3	3	0.909	0.909
8-1	59	9	15	0.868	0.797
8-2	80	4	15	0.952	0.842
8-3	47	0	12	1.000	0.797
9-1	56	11	3	0.836	0.949
9-2	46	22	7	0.676	0.868
9-3	78	3	2	0.963	0.975
Total	945	105	199	0.900	0.826

花数を、目視で計測した結果と、本システムで得られた結果を比較し、ブロックごとの合計値を求め、適合率と再現率を求めた。Table 1 に実験結果を示す。ここで、BN, TP, FP, FN はそれぞれ、目視で確認できた花のうち、撮影地点のブロック番号、計測成功数、誤検知数と同じ花が複数回計測された数、未検知数と過剰な対応付けにより花が計測されなかった数を表す。結果では、平均で適合率は 90.0% と再現率は 82.6% を得た。適合率、再現率共にブロックごとの差が大きく、適合率は最大でブロック 1-2, 8-3 の 1.000, 最小でブロック 9-2 の 0.676, また再現率は、最大でブロック 9-3 の 0.975, 最小でブロック 1-3 の 0.731 を得た。

一株当たりの花数計測について、花数の推移を時系列で表した結果を Fig. 8 に示す。多くのブロックにおいて花数の推移を確認できる。通常的环境下で生育するブロックでは 7 月 22 日以前に花数のピークがあり、薬害を与えるブロックでは、7 月 24 日以降がピークとなるブロックが多いことが確認できる。このように、生育障害の与える影響について、データを取得することが可能である。値が異なっていたブロック 8-1 において、7 月 24 日は花が密集して咲いている部分が、葉によって隠されており、今回の検知領域内で確認することができなかったため、撮影方法の変更とともに検知に利用する範囲と撮影範囲のオーバーラップを新たに定めることで撮影されない花を減少させる。9-1 においては、7 月 27 日に計測結果が目視で確認した花数を上回っている。これは撮影時に対象となる株にフォーカスがあっておらず、画像がボケていたことにより、特徴点抽出の段階で特徴点が得られず、対応付けを行うことができなかったため、一つの花が複数回検出されたためである。

4.3 精度向上の方法

精度の向上を行うためには、花検知の精度の向上が必要である。花検知において、誤検知の原因となっているものは、対象株以外の花が中心となっている。対象

株以外の花については、対象株の背後の株が映り込むことにより起こるため、画像のボケや検知範囲を対象株の茎周辺にのみ制限することで、さらなる精度向上を行う。また未検知の原因は、ブロック 1-1 の 7 月 24 日、ブロック 2-3 の 7 月 22 日、ブロック 8-2 の 7 月 22 日等に多く見られる花の重なりが挙げられる。近い場所に花が咲くことで、一方向から撮影した場合、複数の花が一つの花として検知されている。このため、複数方向から撮影することで、一株の花数を全方向から計測し、重なりを複数方向の画像で補うことで未検知数の減少を行う。

さらなる精度向上の方法として、一画像からの花検知以外に、対応付けの厳密化が挙げられる。現在、花同士が重なり合っている場合、花検知で正確な数を取ることができないことや、花同士の距離が非常に近いことから、対応付けが誤って行われる場合がある。これはブロック 8-2 の 7 月 24 日やブロック 3-2 の 7 月 22 日等に見られた。このため、撮影方法の改良を行い、複数のカメラを利用し同時に撮影を行うことで、単軸ロボット稼働中の農作物の揺れや連続画像撮影時のシャッターの遅れによるカメラ間隔の誤差を軽減し、カメラ間の距離を一定に保つ。また花領域同士が近い場合に対応するため、一画像内で各花領域について自身の周囲の花領域の方向と距離を保持し、連続画像間の対応付けに利用し、対応付けの精度向上を行う。

また、画像がボケている場合、特徴点が得られないことから、撮影時に画像を確認しフォーカスが対象に合っていない場合、再度撮影を行うことで画像の撮影精度の向上を行う。

5 まとめと今後の課題

本論文では、画像処理により農作物の生育状況を観測し収量予測を行うシステム開発を最終目標とし、収量と密接な関係があると考えられる花数計測を行った。実験では領域分割と色相を用いた判定と、領域の統合、識別器を利用し、画像から花領域の検知を行った。また得られた花の連続画像内での対応付けを、特徴点抽出と特徴量の計算と、画像内の花の移動量の推定から行い、一株当たりの花数計測を行った。これら二つの実験において、提案システムでの結果と目視で確認した花との比較を行い、考察を加えた。

実験結果では、一株当たりの花数計測については、平均で適合率は 90%、再現率は 80%以上を得た。ブロックごとにばらつきがあるため、一画像からの花検知手法の精度を高めるとともに、今後、一画像からの花検知の性能向上と、撮影方法を改良することで精度の向上を行うことが可能であると確認できた。また本システムにより、花数の推移を追うことが可能であり、生育障害を与える影響について、データの取得が可能であると確認した。

本研究では連続画像において、一画像からの花検知と一株当たりの花数計測を行ったが、さらに精度向上を行いより詳細な生育情報を得ることが望ましい。また、検知対象となる花を別の農作物に広げることでより価値の高いシステムの構築を目指す。

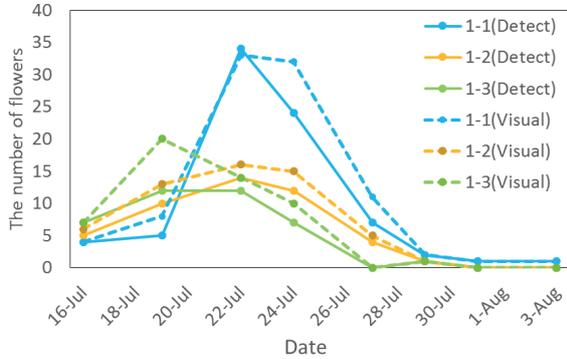
謝辞

本研究は、平成 27 年度農林水産省委託プロジェクト研究「多収阻害要因の診断法及び対策技術の開発」の

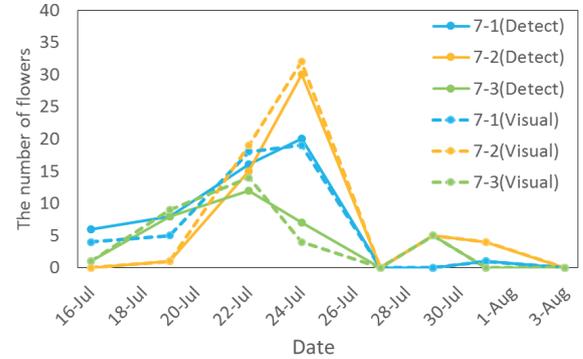
助成を得て行われた。

参考文献

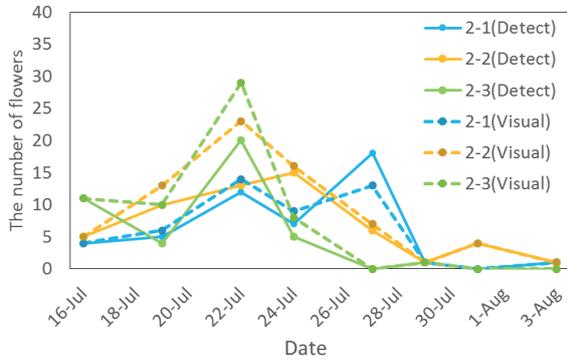
- 1) 農業 ICT ソリューション NEC ホームページ <http://jpn.nec.com/solution/agri/>.
- 2) 食・農クラウド Akisai 富士通ホームページ <http://jp.fujitsu.com/solutions/cloud/agri/>.
- 3) A. Najjar and E. Zagrouba: Flower image segmentation based on color analysis and a supervised evaluation, Proceeding 2nd International Conference ICCIT - 402/406 (2012)
- 4) B. Biradar and S. Shrikhande: Flower Detection and Counting Using Morphological and Segmentation Technique, International Journal of Computer Science and Information Technologies, **6-3**, 2498/2501 (2015)
- 5) R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Sussstrunk: Slic superpixels compared to state-of-the-art superpixel methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, **34-11**, 2274/2282 (2012)
- 6) Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: Gradient-based learning applied to document recognition, Proc. of the IEEE, 2278/2324 (1998)
- 7) E. Rosten, R. Porter, T. Drummond: Faster and Better: A Machine Learning Approach To Corner Detection, Pattern Analysis and Machine Intelligence, 105/119 (2010)
- 8) E. Rublee, V. Rabaud, K. Konolige, G. Bradski: ORB: an efficient alternative to SIFT or SURF, Int. Conference on Computer Vision, 2564/2571 (2011)
- 9) K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman: Return of the Devil in the Details: Delving Deep into Convolutional Nets, British Machine Vision Conference (2014) (arXiv ref. cs1405.3531)
- 10) C. Bishop: Pattern Recognition and Machine Learning, Springer, 1/267 (2010)



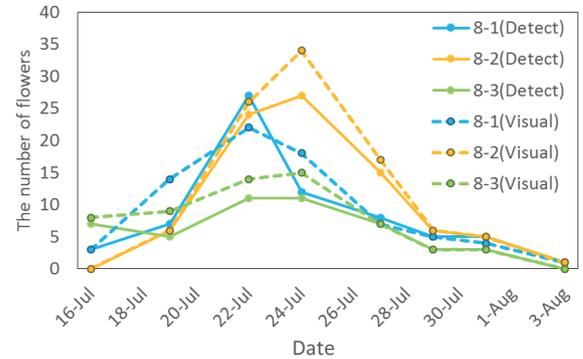
(a) Block1



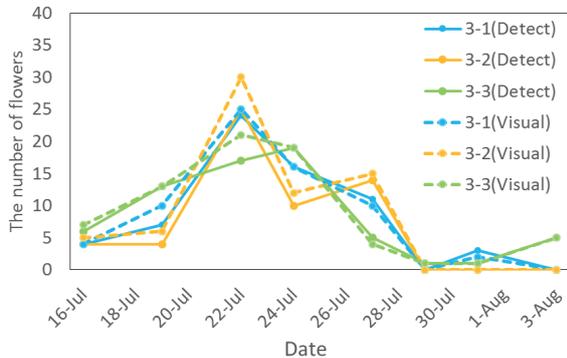
(d) Block7



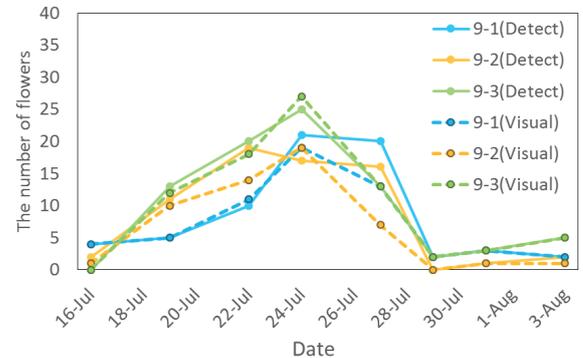
(b) Block2



(e) Block8



(c) Block3



(f) Block9

Fig. 8: The time course of the numbers of flowers. The left column represents results for the blocks where the plants were grown under normal condition (the blocks 1–3), and the right column for the blocks with excess herbicide application (the blocks 7–9). The solid and dashed lines indicate the numbers of flowers detected by the proposed method and the groundtruth counted by human, respectively.

ロボットによる柔軟物操作における複数視点撮影のための ハンドアイシステム

○渡部知俊 松野隆幸 白川智也 見浪護 (岡山大学)

Hand-eye Camera System for Capturing Distance Image from Multi Viewpoints in Deformable Object Manipulation by Robot

*T. Watanabe and T. Matsuno, T. Shirakawa, M. Minami (Okayama University)

Abstract— Our research purpose is manipulation of deformable object by a robot. This paper describes both mechanism of end-effector and view direction of a distance camera. The distance camera is mounted to end-effector to obtain point cloud data of string surface from various viewpoints. In order to obtain low-deficit surface of a string without occlusion, distance images are captured from five different viewpoints. Those five images are merged with ICP algorithm, and improvement of ICP requires information concerning each viewpoints. Therefore, homogeneous transformation matrix is used to decide the photographing direction of the distance camera. At final, the result of string untying experiment is reported.

Key Words: Deformable object, Hand-eye system

1 はじめに

我々の身の回りに存在する布やひもといった物体は柔らかく形状が一定に定まりにくい不定形物体と呼ばれる。この不定形物体のマニピュレーションは様々なに変化する形状に対して柔軟に対応する必要があり、画一的な動作で操作することができないため困難である。近年増加しているセル生産方式ではロボットマニピュレータが作業の自動化に貢献している。このような生産の現場にもケーブルのような不定形物体が存在しており、ロボットによる配線作業の自動化が期待されている¹⁾。また家庭用ロボットが作業を行う人々の生活圏には様々な柔軟物が存在しそれらを自由に操作することは必要不可欠である²⁾。本研究は不定形物体であるひもに着目する。この分野に関する先行研究としては高松ら³⁾によってひもの形状や状態の記述法に関する研究がなされている。また森田ら⁴⁾や若松ら⁵⁾、松野ら⁶⁾はロボットマニピュレータによるひもの形状操作を目的とした研究を行い、カメラで床に置かれたひもの形状を認識し、結び目理論やトポロジーを応用した状態遷移の理論、それらを使ったロボットの動作計画などを提案している。それらに対して我々の研究では、これまで距離カメラで対象となる紐を三次元の点群として取得し、そこから紐の形状を位置ベクトルの列で構成する点連鎖モデルと呼ばれるものを生成するためのアルゴリズムを提案した⁷⁾。このアルゴリズムでは、複数の視点から得られた点群データを合成するICPアルゴリズム⁸⁾、物体表面の法線推定、法線ベクトル情報をもとに紐の中心軸を推定する中心軸推定、紐の太さ情報を利用したマッチング法などの複数の点群処理を施すことによって、紐の交点での誤認識を解消し、オクルージョンのないひもの3次元形状を復元できることを示した。

我々の研究ではひもの形状操作に実現のために Fig.1 に示すように形状認識、形状操作のための計画、ロボットの動作生成、形状操作の実行という4つの段階を設定した。第1段階の形状認識フェイズでは操作の対象となるひもの形状を認識している。認識された形状と

目標形状との差異をフィードバックすることで対象となるひもの形状を制御する。第2段階は形状操作計画の生成フェイズである。ここでは現在のひもの形状から目標形状に変形させるための最適な形状操作を決定する。第3段階の動作生成フェイズでは生成された形状操作計画を実現するためのロボットの動作を生成する。そして第4段階の形状操作実行フェイズでは生成されたロボットの動作を実行する。形状操作が完了されたら再度形状認識フェイズに戻り、目標形状に一致するまでこれらの処理を繰り返す。この際、ロボットによるひもの形状操作を自動化するためにエンドエフェクタに距離カメラを取り付けている。そのためハンドアイシステムを必要とする。本論文では実装されているハンドアイシステムについて述べ、それを用いたひも解き操作実験の結果についても報告する。

2 ひも形状操作システムの概要

本研究で用いているひも形状操作システムのフローチャートを Fig.2 に示す。本システムは形状認識、操作計画の導出、形状操作に大別され、形状認識、操作計画の導出、形状操作の処理を複数回繰り返すことでひもを任意の形状に操作する。

初期条件として目標となるひもの形状が入力され、処理を開始する。最初の処理は形状認識であり、対象となるひもの形状を3次元点群データとして取得する。ひも形状認識法によってひもの位置や形状を表す点連鎖モデルが生成される。次に、ひもの形状を目標形状に近づけるための最適な形状操作の計画を立てる。本研究では、形状操作の基本的な操作としてライデマイスター移動およびクロスを採用している。ライデマイスター移動、クロスは結び目理論で定義された操作であり、それらを組み合わせることであらゆるひもの形状を実現することが証明されている。これらの形状操作の難しさをコストとして定義し、コストが最小となるように形状操作計画を決定する。ロボットの動作生成においては形状操作計画に基づき動作を生成する。最後に生成されたロボットの動作が実行され、再び形状認識から処理を行う。

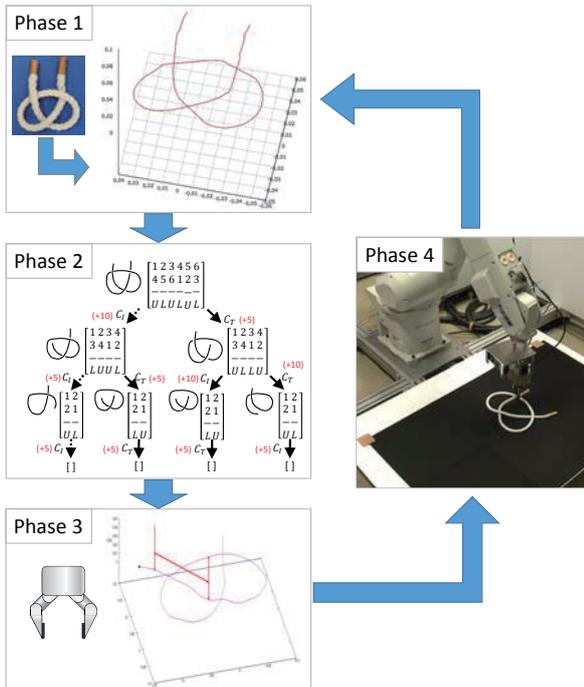


Fig. 1: ひも形状操作の流れ

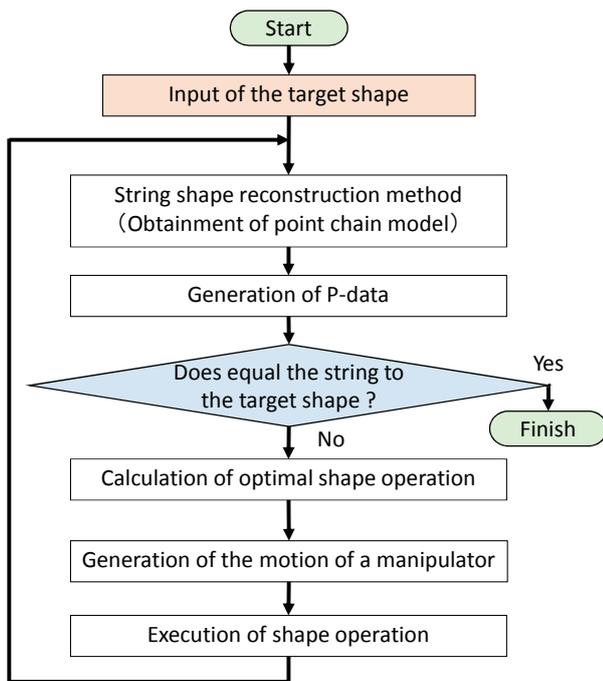


Fig. 2: ひも形状操作システム

2.1 形状認識

形状操作の概要を Fig.3 に示す。形状認識においては、はじめに対象となるひもの表面形状を距離カメラを用いて点群として取得する。この際一視点からの計測ではひもの交差点においてオクルージョンが発生してしまう。そのため異なる5つの視点からひもの表面形状を取得し、ICP アルゴリズムを用いて5つの3次元点群データを合成する。合成された点群データをもとにひもの中心軸を推定し、それらの点を連鎖させることでひもの形状を復元する。ここで生成されたモデ

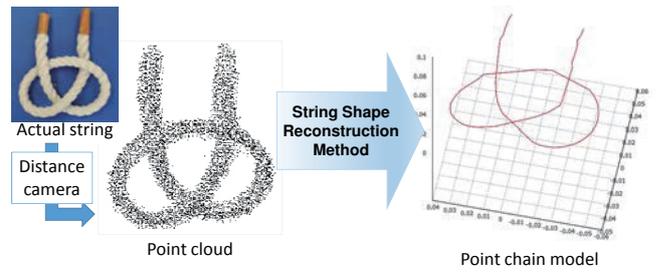


Fig. 3: 形状認識

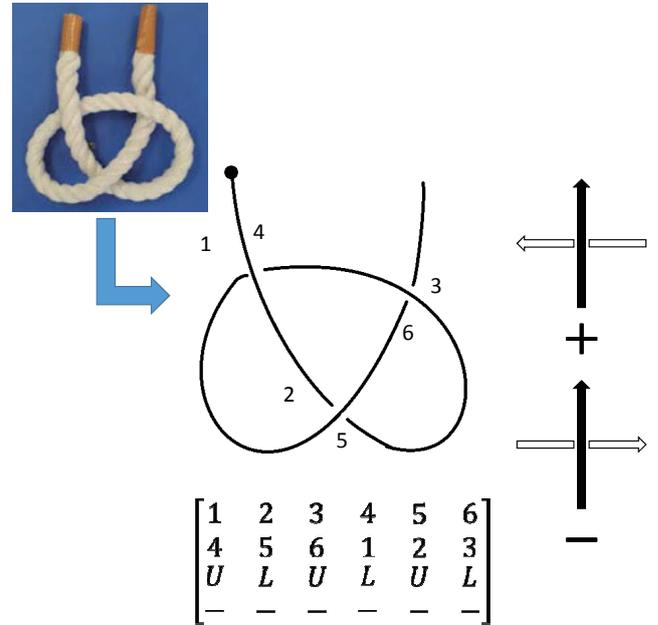


Fig. 4: P-data

ルを点連鎖モデルと呼ぶ。

2.2 形状操作計画の導出

本システムでは結び目理論で定義されているライデマイスター移動およびクロスと呼ばれる操作を複数回行うことでひもを任意の形状に操作を行う。ライデマイスター移動およびクロスの変形を Fig.5 に示す。操作計画においては、点連鎖モデルからトポロジーのみの情報を抽出し Fig.4 のような P-data と呼ばれる行列で表現する。Fig.4 にあるひもの結び目の射影図とその P-data を示す。P-data はひもの射影図から生成される。3次元空間内のひもを2次元平面に射影して射影図として表現する。射影図の中でひもの端を端点と、ひも同士が交差した点を交点と、端点や交点で区切られた線分をセグメントと呼ぶ。P-data 中の第1行および第2行の数字は交点の番号を示しており、一交点には2つの番号が対応している。第3行は交点における交差の方向を表しており、+ または - で表現される。第4行は交点におけるひもの上下方向を表している。U は交点において上側であることを、L は交点において下側であることを示している。

前述したライデマイスター移動およびクロスという操作は P-data に行列の変換という形で行うことが出来、行数を増減させることで処理が行われる。与えられた目標形状と現在の形状を P-data 上で比較しどのよ

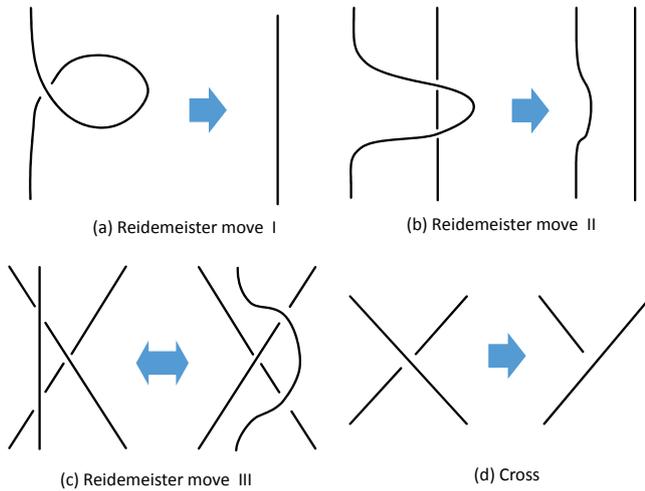


Fig. 5: クロス・ライデマイスター移動

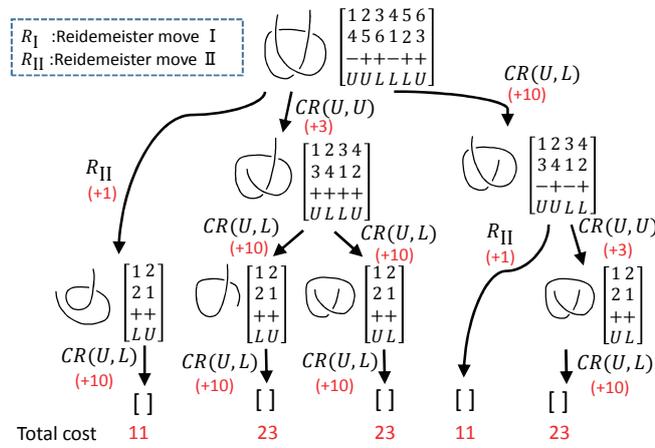


Fig. 6: 操作計画

うな操作が必要か計算し、現在の形状から目標形状までの操作計画を立てる。

2.2.1 ライデマイスター移動・クロス

本研究ではライデマイスター移動およびクロスをひもに対して行う基本的な形状操作として採用する。これらの操作はもともと結び目理論と呼ばれる数学の一分野において定義されたものである。これらの操作には交点を増やす操作と減らす操作が含まれる。本研究ではひもの解きを目的としているため、交点を減らすための操作のみを用いる。ライデマイスター移動にはライデマイスター移動I、ライデマイスター移動II、ライデマイスター移動IIIの3種類が存在する。しかし、ライデマイスター移動IIIでは交点を減らすことが出来ないため、本研究では考慮しない。ライデマイスター移動IはFig.5(a)に示すように単純なループを解く操作である。ライデマイスター移動IIはFig.5(b)のようにあるセグメントが他のセグメントを横切るような操作である。ライデマイスター移動IIIはFig.5(c)のようにひもの一部を交点を越えて動かす動作である。この操作では交点の数が増えないため、本研究では基本的な形状操作として採用していない。クロスはひも全体の両端部分の操作のために考慮された形状操作であり、Fig.5(d)に示すように端点があるセグメントを横切る

操作である。端点には始点と終点の2つが存在する。

2.2.2 最適な形状操作の選択

実際のロボットによるひも解き操作を考えた場合、形状操作の成功率はひもの形状や実行する操作に依存して変化すると考えられる。そこで各基本的な形状操作の難しさを表す指標である”コスト”を導入する。この際にクロスを更に $CR(U,U)$, $CR(U,L)$, $CR(L,U)$, $CR(L,L)$ の4つの形状操作に分類する。UまたはLで示された2つのパラメータは、クロスを行う端点からひもをたどったときの1, 2つ目の交点の上下を表している。

$CR(L,L)$ は形状操作時のロボットハンドとひもとの干渉が大きいことやセグメント同士の摩擦による影響から選択されないこととする。合計5つの基本的な形状操作を交点の減少数、操作の成功率、セグメントとロボットハンドの干渉、セグメント同士の干渉という4つの観点から評価し、ヒューリスティックにコストを決定した。各基本的な形状操作の各項目における評価とコストをTable.1に示す。現在の形状から目標形状に変形するために要したコストの合計を計算する。それらの合計コストを比較し、最小となる経路の最初の形状操作を最適な操作として選択する。交点が1つのとき、例外的に $CR(U,L)$ が選択される。

2.3 形状操作

形状操作においてはP-dataを用いて立てた操作計画に従って操作を行う。しかし、不定形物体のハンドリングによる形状変化を予測することは容易ではないため操作計画の全ての操作を行うのではなく、操作計画の一番目の操作のみを行いその後再度認識過程へ戻る。

3 ハンドアイシステム

ロボットが自立してひもの形状操作を行うために距離カメラをエンドエフェクタに取り付けている。このようなロボットにおいて、カメラ位置とロボットハンドの位置姿勢関係が重要となる。

Fig.7に実験に使用するエンドエフェクタおよびマニピュレータの座標系を示す。ワールド座標系 Σ_w は、マニピュレータのベース部分に固定された座標系 Σ_0 から700[mm]離れた位置に固定する。また、 Σ_e はエンドエフェクタの先端に固定された座標系であり、 Σ_d は点群取得のための距離カメラの座標系である。距離カメラの座標系 Σ_d からエンドエフェクタの先端に固定された座標系 Σ_e までの同次変換行列を式(1)に示す。式(1)の同次変換行列を用いて距離カメラを用いて得られたデータから実際に把持操作するための手先位置を決定することが出来る。

$${}^eT_d = \begin{bmatrix} 1 & 0 & 0 & 100 \\ 0 & 1 & 0 & -13 \\ 0 & 0 & 1 & -208.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

ワールド座標系から見た距離カメラの撮影位置をFig.8に、撮影位置の座標をTable.2に示す。距離カメラはワールド座標系における原点を中心とした半径450[mm]の球面上を移動させる。これらの5つの位置から3次元点群データを取得し、ICPアルゴリズムを用いて合成を行う。ICPアルゴリズムを行うための前

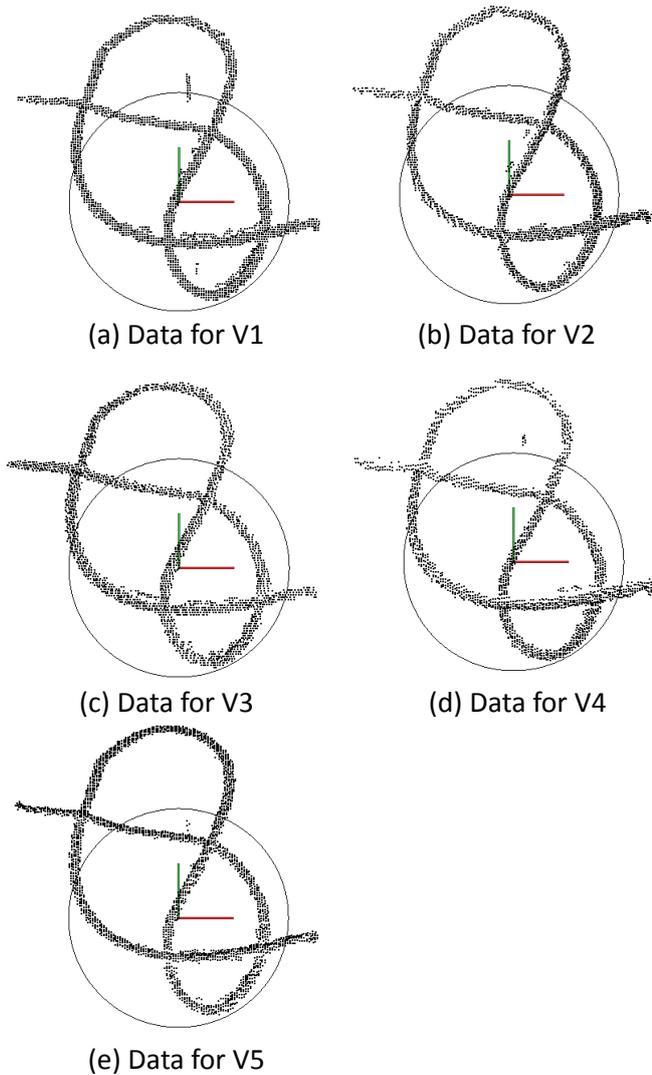


Fig. 9: サンプルデータ

列を用いて初期位置あわせを行う。今回、ICP アルゴリズムは初期位置あわせで出た誤差を修正するために使用する。式 (7) の f が最小となるように \mathbf{R}_i と \mathbf{t}_i を決定する ($i = 2, \dots, 5$)。

$$f(\mathbf{R}_i, \mathbf{t}_i) = \sum |\mathbf{R}_i \mathbf{x}_i + \mathbf{t}_i - \mathbf{y}_i|^2 \quad (7)$$

\mathbf{R}_i は回転行列、 \mathbf{t}_i は並進ベクトル、 \mathbf{x}_i と \mathbf{y}_i は2つの点群データの対応点を表している。ICP アルゴリズムは外れ値に影響されやすいという問題があるが、前処理としてあらかじめ外れ値を除去している。

4 ICP アルゴリズムの有効性検証

ICP アルゴリズムの有効性を検証するためのサンプルデータを Fig.9 に示す。これらの3次元点群データは初期位置あわせを行った後のデータである。

式 (2)~(6) を用いて初期位置あわせを行った際に、誤差が残る。このときの各3次元点群データの座標系を順に Σ_{w1} , Σ_{w2} , Σ_{w3} , Σ_{w4} , Σ_{w5} とする。そのためICP アルゴリズムを用いて初期位置あわせで出た誤差を修正するために用いる。式 (2) を用いて得られた座標系 Σ_{w1} がワールド座標系 Σ_w と一致していると仮定

する。ICP アルゴリズムは式 (2) を用いて得られた座標系 Σ_{w1} を基準とし、式 (7) の f を最小にする回転行列 \mathbf{R} 、並進ベクトル \mathbf{t} を求める。この行列およびベクトルから Σ_{w2} から Σ_{w5} を Σ_{w1} に位置合わせするための同次変換行列 ${}^{w1}T_{Vi}$ が以下のように表される。

$${}^{w1}T_{wi} = \begin{bmatrix} \mathbf{R}_i & \mathbf{t}_i \\ 0 & 1 \end{bmatrix} \quad (8)$$

Fig.9 に示したサンプルデータに ICP アルゴリズムを用いて Σ_{w2} から Σ_{w5} を Σ_{w1} に位置合わせするための同次変換行列を求める。ICP により得られた同次変換行列を用いて座標系 Σ_w から各撮影姿勢までの同次変換行列の推定値 ${}^w\hat{T}_{Vi}$ を式 (9) を用いて求める。式 (9) から得られる Σ_{w1} 座標系から各撮影位置姿勢を表す同次変換行列は式 (10) から (14) のようになる。

$${}^w\hat{T}_{Vi} = {}^wT_{wi} {}^{wi}T_{Vi} \approx {}^{w1}T_{wi} {}^{wi}T_{Vi} \quad (9)$$

$${}^w\hat{T}_{V1} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 450 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$${}^w\hat{T}_{V2} = \begin{bmatrix} -0.26 & 7.27 \times 10^{-9} & 0.97 & -116.5 \\ 1.44 \times 10^{-8} & 1 & 1.16 \times 10^{-8} & -6.5 \times 10^{-6} \\ -0.97 & -1.29 \times 10^{-8} & -0.26 & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$${}^w\hat{T}_{V3} = \begin{bmatrix} 0.26 & -3.57 \times 10^{-8} & 0.97 & 116.5 \\ 2.57 \times 10^{-8} & 1 & 2.98 \times 10^{-8} & -1.2 \times 10^{-5} \\ -0.97 & -2.01 \times 10^{-8} & 0.26 & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$${}^w\hat{T}_{V4} = \begin{bmatrix} -4.04 \times 10^{-10} & -1.38 \times 10^{-8} & 1 & 1.85 \times 10^{-7} \\ 0.26 & 0.97 & -3.5 \times 10^{-8} & -116.5 \\ -0.97 & 0.26 & -8.1 \times 10^{-9} & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$${}^w\hat{T}_{V5} = \begin{bmatrix} -4.70 \times 10^{-6} & -6.3 \times 10^{-7} & 1 & 2.11 \times 10^{-3} \\ -0.26 & 0.97 & -7.2 \times 10^{-7} & 116.5 \\ -0.97 & -0.26 & -4.7 \times 10^{-6} & 434.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

ここで式 (3) から (6) を式 (10) から (13) と比較すると、姿勢行列の最大偏差は 0.004、並進行列の最大偏差は 0.03 であり、ICP による補正は小さい。

ここで ICP アルゴリズムを用いて位置あわせを行った点群と初期位置あわせのみを行った点群を Fig.10 に示す。また、これらの3次元点群データをもとに生成された点連鎖モデルは Fig.11 に示す。Fig.10(a) をもとに生成された点連鎖モデルは Fig.11(a) である。このとき点連鎖モデルは途中で途切れており、正確な形状を認識することが出来ていない。Fig.10(b) をもとに生成された点連鎖モデルは Fig.11(b) である。このときの点連鎖モデルは途中で途切れることなく、全体の形状を認識することが出来ている。また、点群のばらつきを評価するためにひもの半径を求める。Fig.12 に今回サ

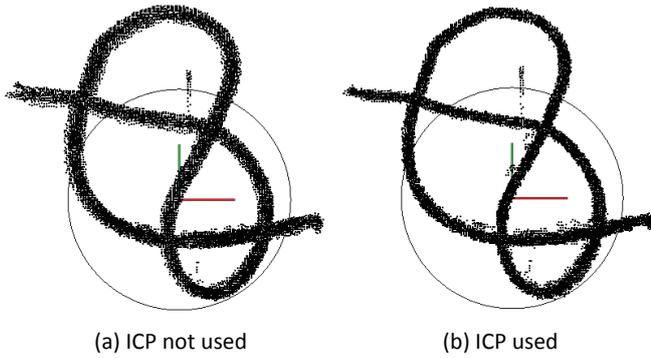


Fig. 10: 点群の比較

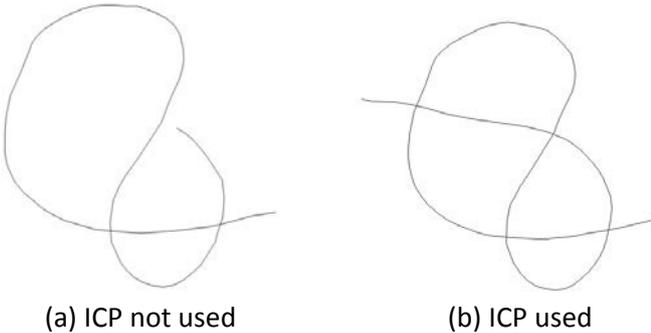


Fig. 11: 点連鎖モデルの比較

サンプリングを行った点を赤い点として示す. それぞれのサンプリング点について, Fig.12をFig.10(a),(b)に投影した際の近傍点における最大半径を求めた結果をTable3に示す. サンプリングした10点中9個の点において, ICPを用いた点群におけるひもの半径がICPが行わなかった場合より小さい. 点連鎖モデルを生成する際に, 点群を細くすることでより精度が高くなるため, ICPアルゴリズムを用いれば補正量は小さいが, 点連鎖モデルを精度よく生成できる.

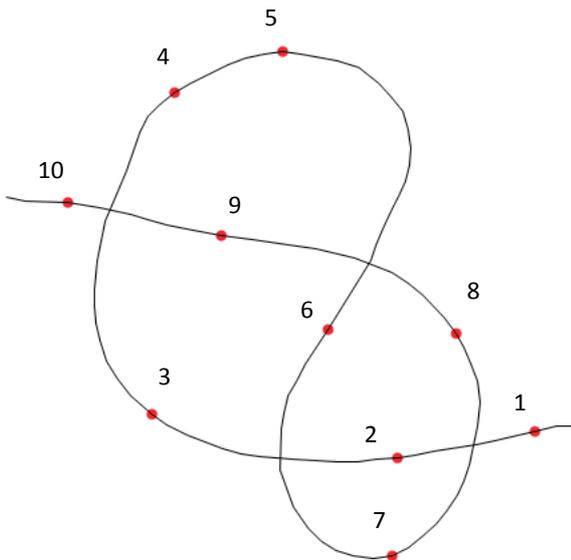


Fig. 12: 半径のサンプリング点

Table 3: 各点における最大半径

Point	maximum radius[mm]	
	ICP	without ICP
No.1	6.27	8.39
No.2	7.52	8.38
No.3	6.39	7.10
No.4	6.60	9.43
No.5	5.53	8.86
No.6	8.69	8.83
No.7	8.96	8.54
No.8	5.44	7.44
No.9	5.77	8.14
No.10	6.32	8.24
平均	6.75	8.34
標準偏差	0.00125	0.00068

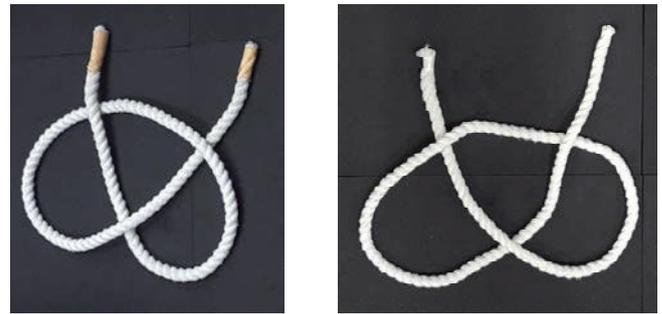


Fig. 13: 実験で用いたひもの形状

5 ひも形状操作実験

今回はひも解き操作の実験について述べる. ひもの交点が0個であるとき, ひもが結ばれていないことは明らかであるため, 本実験における目標形状のP-dataを空行列とする. 今回実験を行ったひもの形状をFig.13に示す. このひもの長さはおよそ1[m]で, 直径は8.4[mm]である. なお, 実験結果についてはひも解き操作の成功ケースのみを報告する.

5.1 実験の流れ

実験の流れをFig.14に示す. はじめにFig.14(a)において異なる5つの位置姿勢から距離カメラを用いて対象となるひもを計測する. その後Fig.14(b)において形状認識を行い, 点連鎖モデルおよびP-dataが生成される. 生成されたP-dataが目標とする形状のP-dataと一致しているか判別をする. 一致していない場合は現在および目標形状のP-dataをもとに現在のひも形状を目標形状に近づけるための操作計画を生成する. さらに, 実際にひもに行う操作の軌道を計算する. Fig.14(c)において実際に操作を行い, Fig.14(d)において操作を実行完了する. これらの処理をひもの形状が目標形状と一致するまで繰り返し行う.

5.2 実験結果

5.2.1 サンプル1

Fig.13(a)に示した形状のひもに対して実験を行った結果をFig.15に示す. また, この形状における操作計画とコストをFig.16に示す.

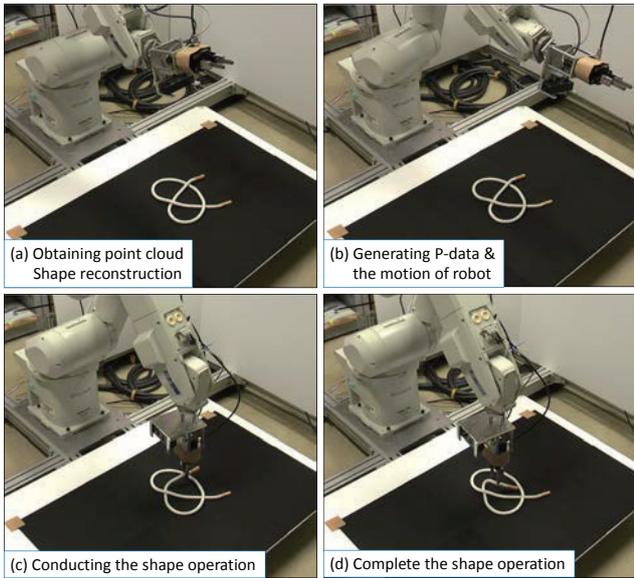


Fig. 14: 実験の流れ

Fig.15(a) に示した初期形状のひもに対して形状認識および操作軌道生成の結果を Fig.15(e) に示す。このとき選択された操作は CR(U,L) である。このとき赤い矢印はロボットによる把持操作の軌道である。操作後のひもの形状は Fig.15(b) である。このとき 2 番の交点においてセグメント同士の干渉が起きたが、交点を取り除くことが出来ている。再度認識および操作軌道の生成を行った結果は Fig.15(f) に示す。このとき選択された操作は CR(L, U) である。操作後のひもの形状は Fig.15(c) である。このときの操作はひも同士の干渉も少なく形状操作は停滞することなく行われた。再度認識および操作軌道の生成を行った結果は Fig.15(g) に示す。このとき交点が 1 個であるため、選択された操作は CR(U,L) であった。操作後の形状は Fig.15(h) となった。このときひもの交点が 0 個となり、ひも解き実験は終了した。

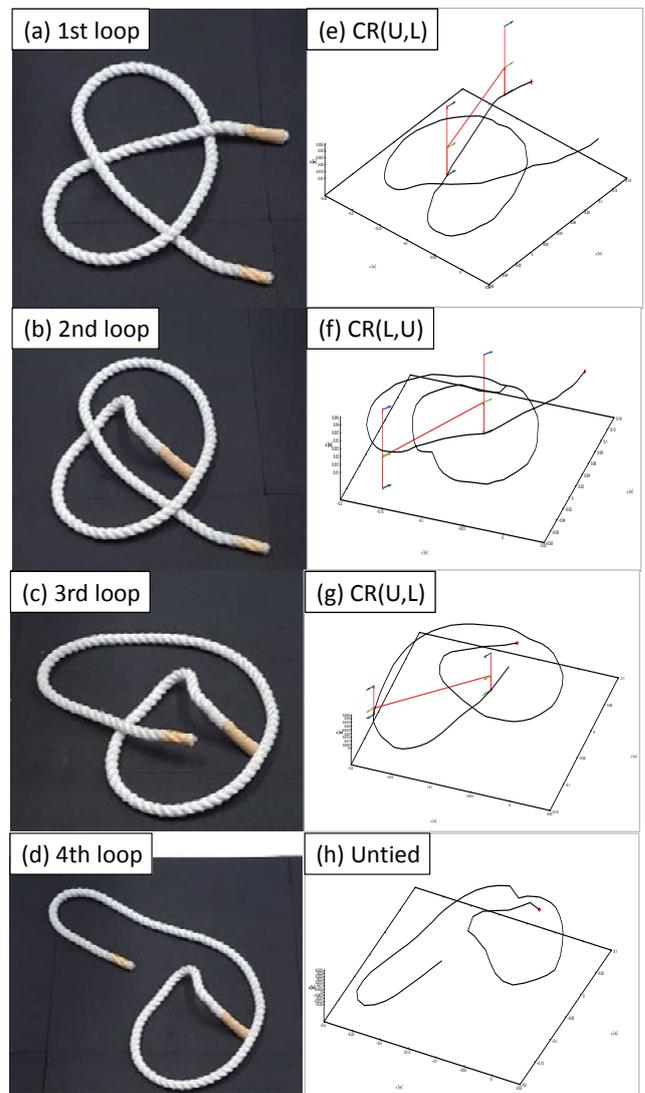


Fig. 15: 実験結果 1

5.2.2 サンプル 2

Fig.13(b) に示した形状のひもに対して実験を行った結果を Fig.17 に示す。Fig.17(a) に示した初期形状のひもに対して形状認識および操作軌道生成の結果を Fig.17(e) に示す。このとき選択された操作は CR(L,U) である。操作後のひもの形状は Fig.17(b) であり、再度認識および操作軌道の生成を行った結果は Fig.17(f) に示す。このとき選択された操作は CR(U,L) である。操作後のひもの形状は Fig.17(c) であり、再度認識および操作軌道の生成を行った結果は Fig.17(g) に示す。このとき交点が 1 個であるため、選択された操作は CR(U,L) であった。操作後の形状は Fig.17(h) となった。このときひもの交点が 0 個となり、ひも解き実験は終了した。サンプル 2 の実験においては全ての動作が停滞することなく行われた。

6 おわりに

本論文では実装されているハンドアイシステムについて述べ、それを用いたひも解き操作実験の結果についても報告した。ひもの形状操作の結果、ひもが距離カメラの撮影範囲外に出てしまうことがあった。現在は

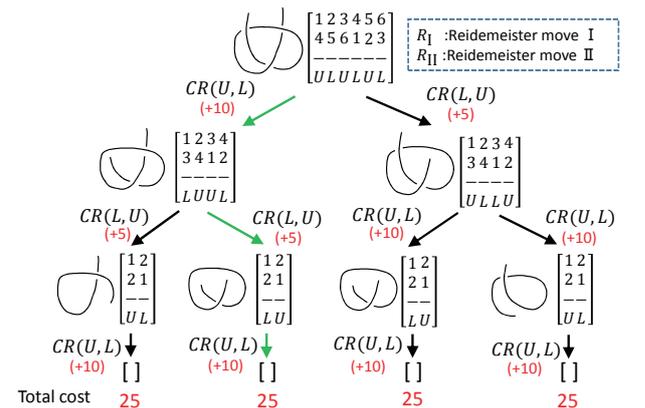


Fig. 16: 実験結果 1-1

あらかじめ指定した 5 つの位置姿勢からのみで距離カメラによる撮影をおこなっており、このような状況ではひもの全体形状を復元することは不可能である。そのためアルゴリズムの改良を行い、任意の位置姿勢から撮影を行い点群データを補完する必要があると考えられる。

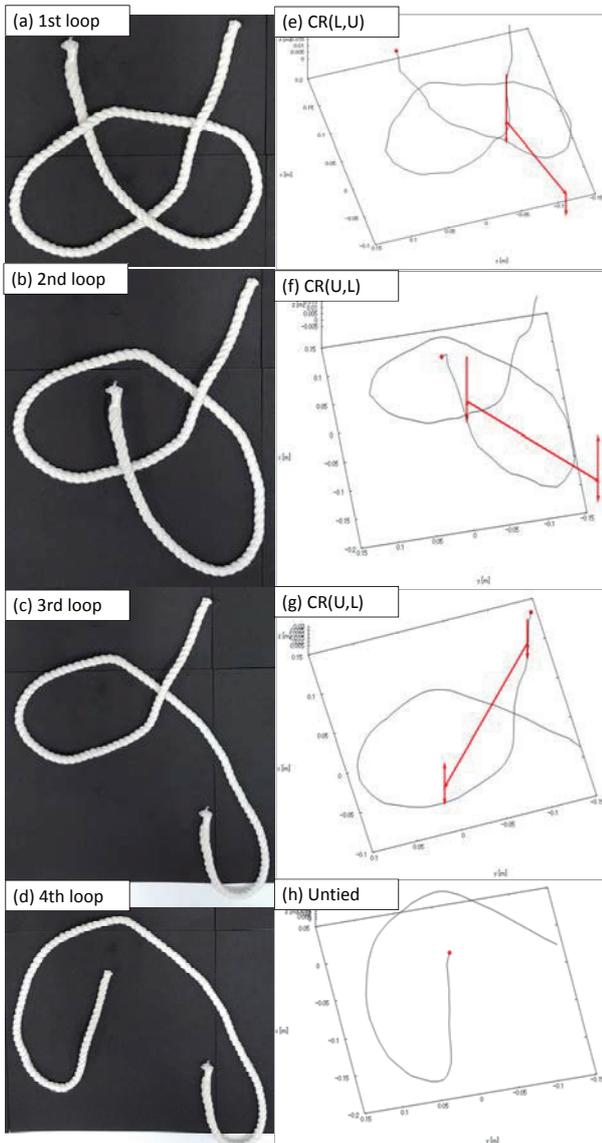


Fig. 17: 実験結果 2

のためのロープの形状認識”, 計測自動制御学会論文集, Vol.41, No.4, p.366~p.372, 2005.

- 7) Keisuke Mukai, Takayuki Matsuno, Akira Yanou, Mamoru Minami, "Shape Modeling of A String And Recognition Using Distance Sensor", IEEE 24th International Symposium on Robot and Human Interactive Communication, Aug.31-Sep.4, 2015.
- 8) 中井康博, 美濃導彦, "複数の点群に対する位置合わせ手法の性能比較", 特別研究報告書, 2012.

謝辞

本研究は MEXT 科研費 JP26820088 の助成を受けたものです。

参考文献

- 1) 鷺見和彦, "解説 柔軟物も取り扱える生産用ロボットシステムの開発," 日本ロボット学会誌, Vol.27, No.10, pp1082-pp1085, Jul.2009.
- 2) 金子学, 柿倉正義, "家庭用サービスロボットのための柔軟物体のハンドリングに関する研究-布地の画像処理について-", コンピュータビジョンとイメージメディア, 135-2, 2002.
- 3) 高松淳, 森田拓磨, 小川原光一, 木村浩, 池内克史, "ロボットによる実行を目的とした紐結び作業の記述", 日本ロボット学会誌 Vol.23No.5, pp.572~582, 2005.
- 4) 森田拓磨, 高松淳, 小川原光一, 木村浩, 池内克史, "観察によるひも結び動作の学習", コンピュータビジョンとイメージメディア, 135-12, 2002.
- 5) 若松栄史, 妻屋彰, 荒井栄司, 平井慎一, "結び/解き操作を含めた線状物体のマニピュレーション", 日本ロボット学会誌, Vol.23, No.3, pp.344~351, 2005.
- 6) 松野隆幸, 玉置大地, 新井史人, 福田敏男, "トポロジカルモデルと結び目不変量を用いたマニピュレーション

組み込みシステム向け異常検知方式

○中川慎二 (株式会社 日立製作所)

Anomaly Detection Method for an Embedded System

* S. Nakagawa (Hitachi, Ltd.)

Abstract— An anomaly detection using a machine learning has attracted attention recently. Although a conventional anomaly detection method has been applied to a system with a comparatively large scale, demand for an anomaly detection at an edge increase because of spreading of IoT (Internet of Things), advancement of processing at an edge such an automated vehicle. In this presentation, an anomaly detection method for an embedded control system is proposed. Clustering is operated using normal data, and the space constituted by each cluster is approximated by a hyper-rectangular. Combining each hyper-rectangular, whole normal space is composed. When a new sample to be determined whether it is normal or anomaly is obtained, an outlier anomaly detection and an transition anomaly detection operate. The outlier anomaly detection operates based on whether the sample is within the hyper rectangular or not, which is corresponded to a center vector the sample is closest to in terms of distance. The transition anomaly detection operates using the transition pattern of the center vector. Since the normal space is represented by a hyper-rectangular in the method, calculation cost for detecting an anomaly is lower, and it is suitable for an embedded system. It is possible to optimize the outlier anomaly detection accuracy and the transition anomaly detection accuracy according to the number of hyper-rectangular because increasing the number of hyper-rectangular make the approximation accuracy of the normal space higher. Results of evaluation for the anomaly detection method using a real system are also shown.

Key Words: Anomaly detection, Embedded system, Clustering, Hyper-rectangular-

1 はじめに

機械学習を用いた異常検知が、近年、注目されている¹⁻³⁾。従来の異常検知方式は、比較的規模の大きなシステムに適用されてきたが、今後、IoT(Internet of Things)の普及、自動運転車に代表される端末での処理高度化などを背景に、エッジでの異常検知のニーズが高まる。

本発表では、組み込み制御システムへの適用を想定した異常検知方式を提案する。正常データを用いて、クラスタリングを行い、各クラスタで構成される空間を超直方体で近似表現する。各超直方体を組み合わせて、正常空間全体を近似表現する。新たな検知対象のサンプルが得られたとき、サンプルが距離の意味で最も近い中心ベクトルに対応する超直方体内にサンプルがあるか否かで外れ値異常検知を行う。また、サンプルが距離の意味で最も近い中心ベクトルでサンプルを代表し、中心ベクトルの遷移パターンを用いて、サンプルの遷移異常を簡易的に検出する。

本方式では、正常空間を超直方体で表現するので、異常検知時の計算負荷が少なく、組み込みシステムに適する。また、クラスタ(超直方体)の数を増やすことで、正常空間の近似精度は上がるので、外れ値異常検知精度、遷移異常検知精度もクラスタ数の増加に応じて最適化することが可能である。実システムを対象に本異常検知方式を評価した結果についても述べる。

2 アプローチ

本研究の目的は、主に組み込み制御システムの動作異常を検知することである。下記を前提とした。

- ・過去に実績のあるデータを、正常データとする。
- ・過去に実績のないデータは、すべて異常データとする。

上記の前提の下、制御システムの動作異常を下記の2つに分類した。

- ・外れ値異常(outlier anomaly)

正常時に期待される値から外れた状態を指す。従来からある異常検知対象の一つである。

- ・遷移異常(transition anomaly)
正常時に期待される値の遷移から外れた状態を指す。上記2つの異常を検知するアプローチを Fig.1 に示す。

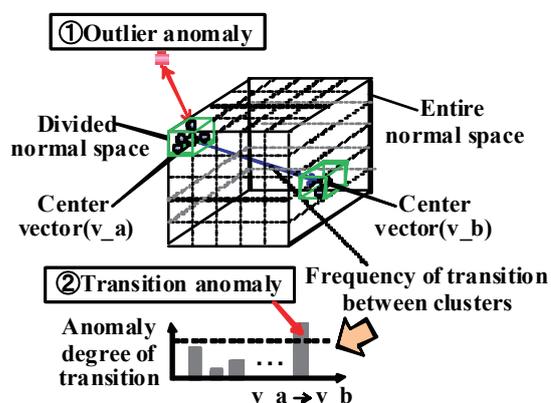


Fig.1: Approach

①外れ値異常の検知

- ・正常動作時における制御に関するパラメータを要素とするベクトル(以下、特徴ベクトル)が、存在するベクトル空間を正常空間とする。
 - ・異常検知対象である特徴ベクトルと正常空間からの距離の大きさに基づいて、外れ値異常判定をする。
- なお、本発表では、異常検知対象である特徴ベクトルが、正常空間内にあるか否かで、外れ値検知を行う。

②遷移異常の検知

- ・正常空間を分割し、異常検知対象であるベクトルがどの分割された正常空間に属するかを決定する。
- ・検知対象ベクトルが属する分割正常空間もしくは分割正常空間の代表ベクトル間の遷移で、検知対象ベクトルの遷移を簡易的に代替表現し、遷移発生の異

常度の大きさに基づいて、遷移異常判定をする。本発表では、異常検知対象である代表ベクトルの遷移発生頻度が0より大きいとき、正常と判定し、0のときは遷移異常と判定する。

3 訓練フェーズ

3.1 正常空間の構築

一般に異常検知は、その他の機械学習と同様に、訓練フェーズとテストフェーズの2つのフェーズに分かれる。訓練フェーズで、正常空間を学習する。テストフェーズで学習した正常空間を用いて検知対象ベクトルの異常を検知する。

訓練フェーズの正常空間の生成方法について述べる。

- i) ベースクラスタリング
クラスタリング手法 k-means を用いて、クラスタリングを行う⁴⁾。クラスタリング数は、目標の異常検知性能および正常検知性能が得られるように決める。k-means の中心ベクトルの初期値は、k-means++法で決める⁵⁾。
- ii) 分割正常空間の生成
各クラスターに属する特徴ベクトルの各要素の最小値および最大値で構成される超直方体を分割正常空間とする。
- iii) 正常空間全体の生成
超直方体により構成される超立体の内部を正常空間とする。

Fig.2 に、本方式による3次元の場合の正常空間のイメージを示している。

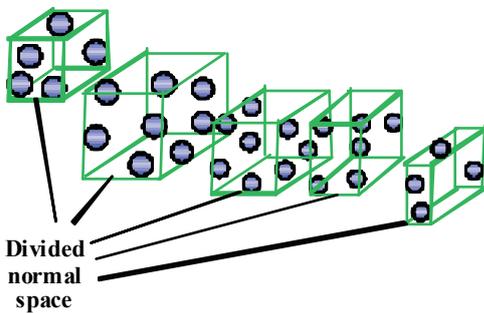


Fig.2: Image of normal space

Fig.3 に、特徴ベクトルが2次元の場合の k-means によるクラスタリング結果の例を示す。

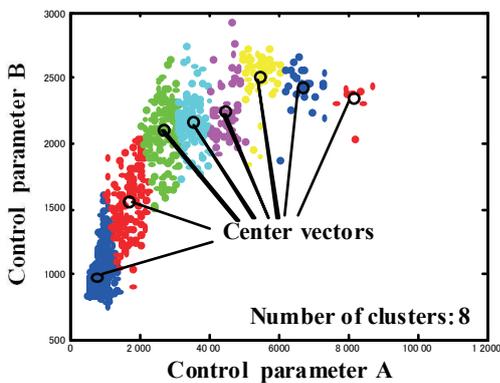


Fig.3: Base clustering

Fig.4 に超直方体の組み合わせによる正常空間生成の例を示す。

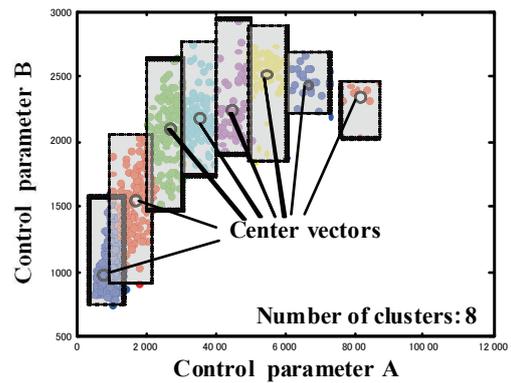


Fig.4: Making divided normal space and entire normal space

3.2 遷移発生頻度の演算

訓練時の遷移パターン発生頻度の演算方法について述べる。特徴ベクトルの時系列順の遷移の発生頻度を、特徴ベクトルが属する分割正常空間の間の遷移の発生頻度で代替演算する。Fig.5 は、特徴ベクトルが2次元の場合の特徴ベクトルの遷移の様子(上側)と分割正常空間の間の遷移発生頻度のイメージ(下側)を示している。

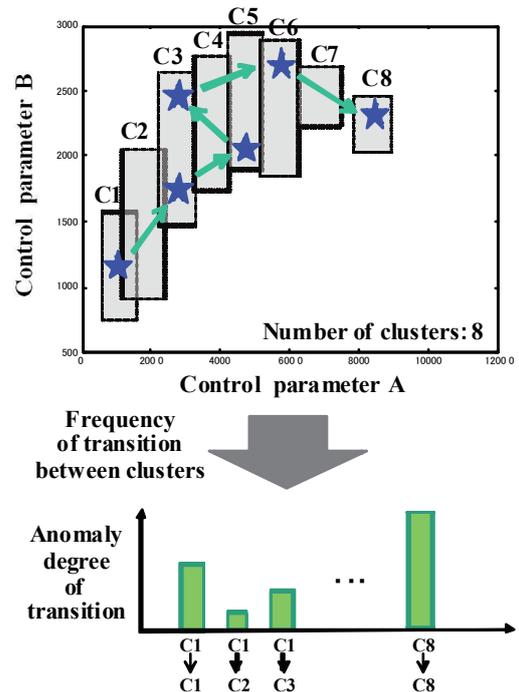


Fig.5: Calculating frequency of transition

4 テストフェーズ

4.1 外れ値異常

テストフェーズの処理について述べる。異常検知対象の特徴ベクトルを時系列順に下記処理を実施する。

- i) 検知対象の特徴ベクトルが得られたら、L2距離が、もっとも近いクラスタの中心ベクトルを選ぶ。
- ii) 選んだ中心ベクトルに対応する分割正常空間の内部に、検知対象の特徴ベクトルが存在しなければ、外れ値異常と判定する。

Fig.6は、外れ値異常の検知例を示している。青色の★は、L2距離がもっとも近い中心ベクトルに対応する分割正常空間C7の内部にあるので正常値と判定する。赤色の★は、L2距離がもっとも近い中心ベクトルに対応する分割正常空間C7の内部にないので、外れ値と判定する。

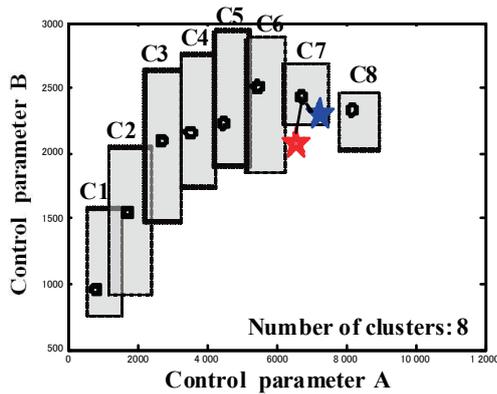


Fig.6: Outlier anomaly detection

4.2 遷移異常

- i) 「新しい特徴ベクトルが所属する分割正常空間（中心ベクトル）」→「前回の特徴ベクトルが所属する分割正常空間（中心ベクトル）」の遷移頻度が0のとき、遷移異常と判定する。
- ii) 外れ値異常発生時は、遷移異常であるのは、自明であるので、外れ値異常判定を優先させる。
- iii) 「前回の特徴ベクトルが外れ値」で、「今回の特徴ベクトルが分割正常空間内部に存在する」ときは、遷移異常とする。今回の特徴ベクトルは、外れ値ではなく、外れ値から正常値への遷移は、訓練時には発生していないことを意味するため、遷移異常と判定する。

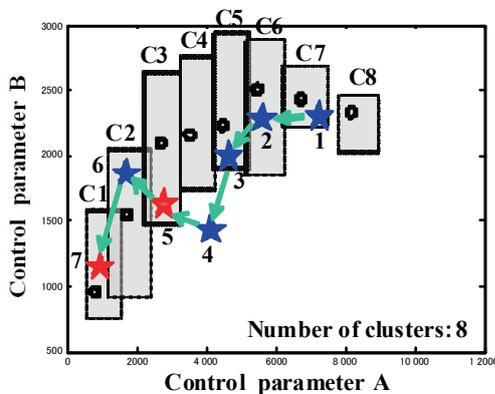


Fig.7: Transition anomaly detection

Fig.7は、遷移異常の例を示している。特徴ベクトルの（番号4）から特徴ベクトル（番号5）に遷移した場合が遷移異常である。また、特徴ベクトル（番号6）から特徴ベクトル（番号7）の場合は、正常空間内での遷移ではあるが、本例においては、この遷移は訓練フェーズでは発生していない場合（発生頻度が0の場合）を示しており、遷移異常と判定する。

3.3 本方式の特徴

本方式による異常検知は、下記の特性を持つ。

- i) テストフェーズの処理が簡便となり、組み込みシステムに向く。

- 外れ値異常検知

各クラスタに属する正常ベクトルの各要素の最小値および最大値で囲まれる超直方体を分割正常空間とするので、異常検知対象の特徴ベクトルが、正常空間の内部にあるか外部にあるか外れ値異常検知処理が簡便となる。

- 遷移異常検知

異常検知対象ベクトルの遷移頻度を分割正常空間（中心ベクトル）の間の遷移頻度で簡易的に代替するので、遷移ケースの組み合わせ爆発を抑制できる。

• なお、訓練フェーズは、クラスタリング処理を行うので、必ずしも組み込みシステムには向かない。サーバーなどでの集中処理が考えられる。

- ii) 正常標本精度と異常標本精度の双方を向上させることが可能。

訓練フェーズで用いた正常ベクトルは、いずれかの分割正常空間内に存在するため、異常検知対象の特徴ベクトルが訓練時に用いたベクトルのいずれかと一致する場合は、必ず正常と判定され、異常と誤判定されることはない。

また、分割正常空間を組み合わせることで正常空間の全体を構成するので、分割正常空間の数を多くするほど、訓練フェーズで用いた正常ベクトルで構成される（真の）正常空間に近づき、（真の）正常空間以外の空間は、（真の）異常空間に近づく。したがって、分割正常空間の数を多くすることで、正常標本精度と異常標本精度の双方を向上させることが可能である。

- iii) 目標性能、コンピューティングパワーに応じた性能設計が可能。

分割正常空間の数で検知精度を調整することが可能であり、目標性能を満足する最小の分割正常空間を選べば、計算リソースの消費を最小限に抑えることができる。

5 実験結果

5.1 実験設備と訓練用データ

小型の自律走行体の制御システムを対象とした。異常検知対象データは、自律走行体の制御操作量である目標車速と目標回転角速度とした。Fig.7に、訓練データ（約360回分の走行データ）を示す。

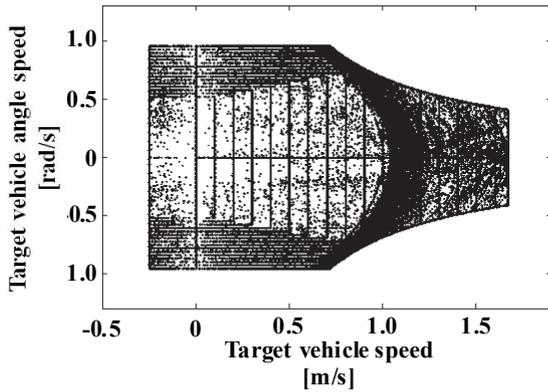


Fig.8 Sample data for training

5.3 実験結果

Fig.9 に、分割正常空間の数が 2 ~ 500 のときの正常空間を示している。分割正常空間の数が増えるほど、訓練データの分布の近似精度が上がっていることがわかる。

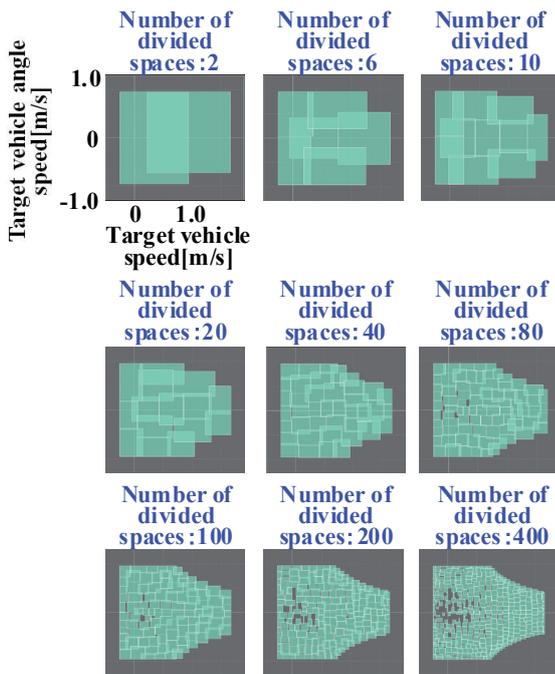


Fig.9 Normal space comprising divided spaces

Fig.10 は、分割空間を 400 としたときの、走行データの走行回数に対する正常空間を示している。走行回数が増えるに応じて、正常空間が拡大していることがわかる。

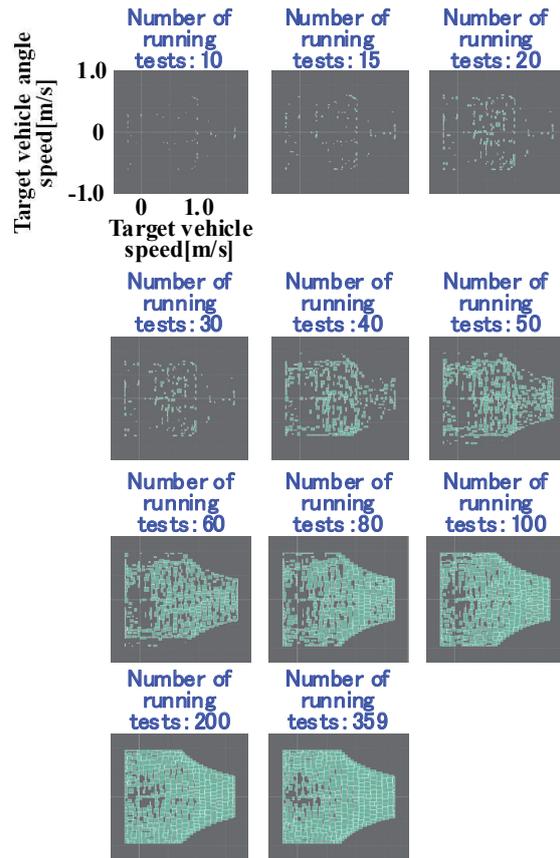


Fig.10 Normal space comprising divided spaces

Fig.11 に外れ値異常検知精度の評価に用いた正常データと異常データを示している。正常データは、訓練データに用いていない正常走行時のデータを用いた。外れ値異常のデータは、Fig.11 に示される赤の部分であり、人工的に生成した。

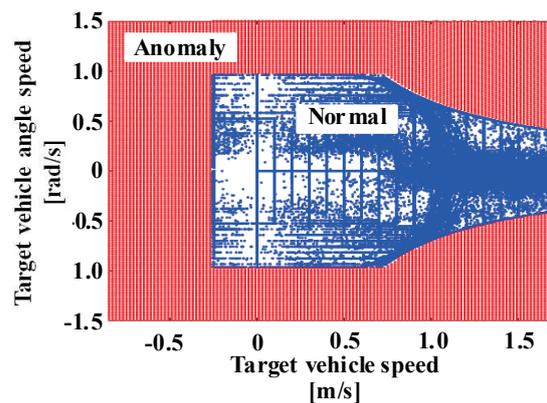


Fig.11 normal and anomaly sample values for outlier anomaly

遷移異常検知精度に用いた正常データは、Fig.11 に示すデータを用いた。Fig.11 に示す正常データにおいて、正常データにはない未発生の変移パターンを人工的に生成し、遷移異常データとした。

Fig.12 は、分割正常空間数に対する外れ値異常検知

の正常標本精度（正常データを正常と判定する率）と異常標本精度（異常データを異常と判定する率）を示している。正常標本精度は、すべての分割正常空間数で、ほぼ100%となった。また、異常標本精度は、分割正常空間数2~100のときは、90~100%で単調増加し、分割正常空間数が200以上のときは、ほぼ100%となった。以上、分割正常空間数が200以上のとき、Fig.11に示す正常データに対する正常標本精度と外れ値異常データに対する異常標本精度の双方がほぼ100%となった。

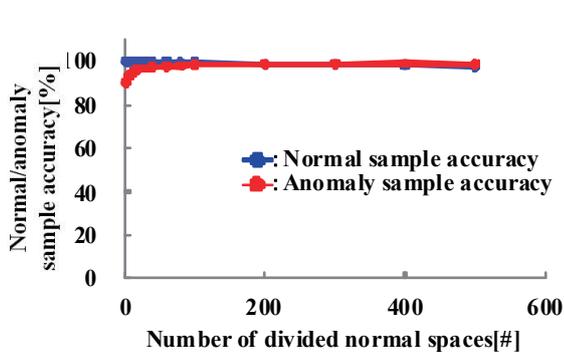


Fig.12 relation between number of divided normal spaces and normal/anomaly sample accuracy on detecting outlier anomaly

Fig.13は、分割正常空間数に対する遷移異常検知の正常標本精度と異常標本精度を示している。正常標本精度は、分割正常空間数が増えるにつれ、やや減少する傾向となったが、いずれも90%以上であった。異常標本精度は、分割正常空間数が増えるにつれ、増加する傾向となり、分割正常空間数400と500で、異常標本精度は、85%以上となった。

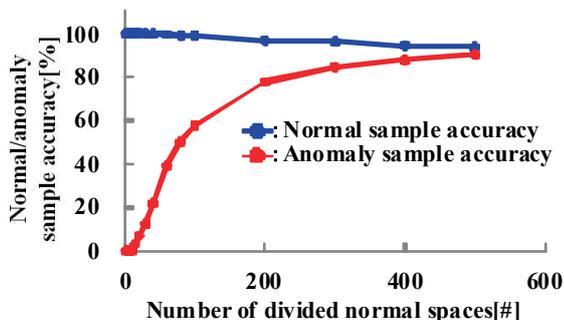


Fig.13 relation between number of divided normal spaces and normal/anomaly sample accuracy on detecting transition anomaly

Fig.14は、分割正常空間数が400のときの、訓練データに用いた走行データのケース数（走行回数）に対する外れ値異常検知の正常標本精度と異常標本精度を示している。

・正常標本精度

走行データのケース数が1~100のときは、ケース数増加に伴い、正常標本精度が単調増加した。ケース数増加に伴い、正常空間が成長し、それに連れて、正常データを正しく検知できるようになったと考える。ケース数が100で、正常標本精度は、約

96%となり、以降は、微増の傾向となった。正常空間が十分に成長したためと考える。

・異常標本精度

走行データのケース数1~500の全ケースで、異常標本精度は、99%以上であった。ケース数が増える正常空間は広がるが、ケース数に関わらず、Fig.11に示す異常データのほとんどが、訓練により得られる正常空間の外側にあったため、異常データを正しく異常判定したと考える。

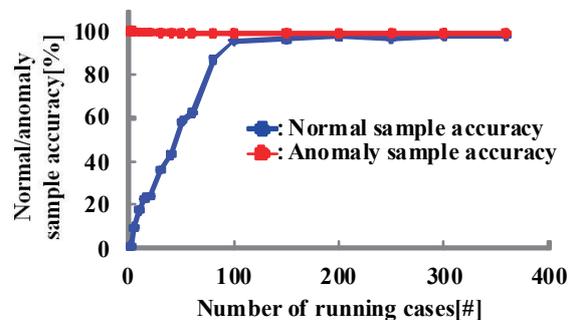


Fig.14 relation between number of learning cases and normal/anomaly sample accuracy on detecting outlier anomaly

Fig.15は、分割正常空間数が400のときの、訓練データに用いた走行データのケース数（走行回数）に対する遷移異常検知の正常標本精度と異常標本精度を示している。

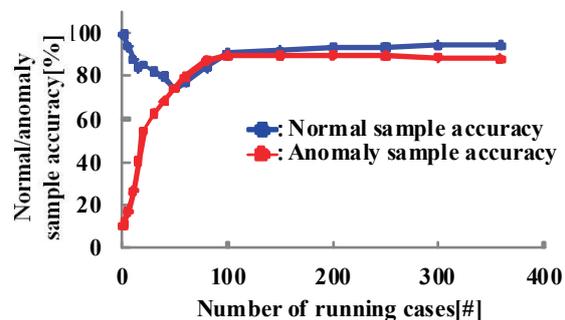


Fig.15 relation between number of learning cases and normal/anomaly sample accuracy on detecting transition anomaly

・正常標本精度

走行データのケース数が1~50のときは、ケース数増加に伴い、正常標本精度が単調減少した。ケース数が60以上では、ケース増加に伴い、正常標本精度は、単調増加した。前述したように、遷移異常は、外れ値から分割正常空間への遷移時もしくは分割正常空間から分割正常空間への遷移に実績がないときに発生する方式としている。同じく前述したように、ケース数増加に伴い、正常空間が成長していくが、ケース数が1~50のときは、Fig.9における正常空間成長の様子およびFig.14における外れ値異常検知の正常標本精度が示すように、訓練で得られる正常空間が未発達である。そのため、テストデータのほとんどが外れ値異常と判定され、結果として、遷移異常判定数が減ったためと考える。すなわち、

ケース数が1～50においても、正常遷移パターンの学習は進んでいるが、外れ値異常判定が非常に多いため、その効果が現れていないと考える。ケース数が60以上では、Fig.14で示すように、外れ値異常検知の正常標本精度が高くなり（外れ値異常判定が減り）、そのため、正常遷移パターンの学習が進んだ効果が現れてきていると考える。ケース数1000のとき、正常標本精度は85%を越え、以降も微増し、ケース数359のとき、約94%となった。

・異常標本精度

走行データのケース数が1～100のときは、ケース数増加に伴い、異常標本精度が単調増加し、ケース数80のとき、異常標本精度は目標の85%を越えた。上述したように、ケース数増加に伴い、正常空間が成長していくが、ケース数が少ないときは、テストデータのほとんどが外れ値異常と判定され、遷移異常判定数が減るので、遷移異常の異常標本精度は低いと考える。ケース数が増えるにつれ、正常空間が発達することで、外れ値異常判定が少なくなり、また、遷移パターンの学習が進むため、異常標本精度は高くなっていくと考える。ケース数100以上では、異常標本精度は、約90%でサチュレートしている。さらに精度を上げるためには、Fig.13で示すように、分割空間数を多くする必要がある。

6 おわりに

(1) 組み込み制御システムへの適用を想定した異常検知方式を提案した。

- ・過去に実績のあるデータを正常データとし、過去に実績のないデータは、すべて異常データとすることを前提とした。
- ・上記前提の下、制御システムの外れ値異常と遷移異常を検知する方式として、検知対象である制御パラメータを要素とする特徴ベクトルが正常時に存在する正常空間を超直方体の組み合わせで構成し、特徴ベクトルが超直方体内にあるか否かで外れ値異常を検知し、特徴ベクトルが属する超直方体間の遷移頻度で遷移異常を検知する方式とした。
- ・本方式では、テストフェーズの処理が簡便となり組み込みシステムに向く。訓練フェーズは、クラスタリング処理を行うので、必ずしも組み込みシステムには向かない。

(2) 実システムである小型の自律走行体の制御システムを対象に本異常検知方式を評価した。

- ・自律走行体の目標車速と目標回転角速度を対象に性能を評価した。
- ・外れ値異常検知の性能評価においては、分割正常空間数が200以上のとき、正常標本精度と異常標本精度の双方がほぼ100%となった。
- ・遷移異常検知の性能評価においては、分割正常空間数が400と500のとき、正常標本精度と異常標本精度の双方が85%以上となった。
- ・分割正常空間数が400のとき、走行データのケース数（走行回数）を増やすほど、外れ値異常検知および遷移異常検知の双方において、正常標本精度と異常標本精度は共に向上した。

(3) 今後の課題について述べる。

- ・検知パラメータの選び方
本実験では、自律走行体の目標車速と目標回転角

速度を対象とした。この2つのパラメータのみで動作を決めていることが明らかだったためである。しかし、制御システムが複雑化した場合、制御システムの仕様がすべて明らかでない場合など、検知パラメータの選択も課題となる。

・正常データの与え方

本手法は、実績のあるデータを正常データと定義し、実績に応じて正常領域が広がる。正常領域で、制御システムの動作範囲が決まるので、初期状態は、一定の広がりのある正常空間を与える必要がある。市場投入前のオフライン検証データを初期正常空間として用いることが考えられる。また、市場投入後、正常空間を広げていくためには、実績データの収集が必要である。

・異常データの与え方

異常検知精度は、異常標本がないと評価できない。一般に、(正常標本数) > (異常標本数)であり、異常を実際に起こすことは、現実的には困難なことが多い。本手法では、正常データ(実績のあるデータ)以外は、すべて異常データとし、正常空間を構築する。異常検知精度の評価においては、人工データを作成した。

参考文献

- 1) Sugiyama, M., et.al.: A density-ratio framework for statistical data processing, IPSJ Transactions on Computer Vision and Applications, (2009)
- 2) Takimoto, M., et.al.: Visual Inspection of Precision Instruments by Least-Squares Outlier Detection, データマイニングと統計数理研究会(第10回), (2009)
- 3) 野田, 他: 高度予兆診断システムの開発, 日本機械学会動力・エネルギー技術シンポジウム講演論文集, 16, 39/42 (2011)
- 4) MacQueen, J. B.: Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 281/297, (1967)
- 5) David Arthur: k-means++: The advantages of careful seeding, Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithm, 1027/1035, (2007).

カオス四元数連想メモリによる動的な想起

○ 長名優子 (東京工科大学)

Dynamic Association in Chaotic Quaternionic Associative Memory

*Y. Osana (Tokyo University of Technology)

Abstract— In this report, we investigate a dynamic association ability of the Chaotic Quaternionic Associative Memory (CQAM). The Chaotic Quaternionic Associative Memory is based on the Hopfield Network and composed of Chaotic Quaternionic neuron models. It can deal with multi-valued patterns. Although the Chaotic Quaternionic Associative Memory can realize dynamic association of stored multi-valued patterns, its ability is very sensitive to parameters of the Chaotic Quaternionic neuron models. We carried out a series of computer experiments and confirmed that the Chaotic Quaternionic Associative Memory needs enough time to recall plural stored patterns.

Key Words: Chaotic Quaternionic Neuron, Dynamic Association

1 はじめに

生物の神経系において見られるような柔軟な情報処理を行う手法として、ニューラルネットワークの研究が盛んに行われており、様々な連想記憶モデルが提案されている^{1)–5)}。しかし、これらの連想メモリの多くは2値パターンを扱うことを前提としており、多値パターンを扱うことはできない。多値パターンを扱うことができるモデルとしては、複素ニューロンモデル⁶⁾⁷⁾や四元数ニューロンモデル⁸⁾が提案されている。これらのモデルは、ニューロンの入出力や内部状態として複素数値や四元数値を扱えるように拡張されたモデルであり、複素数値や四元数値を多値に割り当てることで、多値パターンを扱うことができる^{9)–12)}。

また、一方で、柔軟な情報処理を行う手法の一つとしてカオスが注目されている。カオスは、決定論的な時間発展をする非線形なシステムに生じる長期予測不可能な現象であり、生物の脳や神経系でも観測されており、記憶や学習においても重要な役割を果たしていると考えられている¹³⁾。人工的なニューラルネットワークにおいても、実際の神経細胞でみられる時空間加算、不応性、連続値出力などを考慮することでカオスを導入したカオスニューロンモデル¹⁴⁾が提案されている。また、カオスニューロンモデルから構成される連想メモリでは、記憶したパターンを動的に想起できることが知られている¹⁴⁾¹⁵⁾。

そのような中で、複素ニューロンモデル⁶⁾⁷⁾とカオスニューロンモデル¹⁴⁾を組み合わせたカオス複素ニューロンモデル¹⁶⁾¹⁷⁾や四元数ニューロンモデル⁸⁾とカオスニューロンモデルを組み合わせたカオス四元数ニューロンモデル¹⁸⁾も提案されている。カオス複素ニューロンモデルから構成されるカオス複素連想メモリ¹⁶⁾¹⁷⁾では、多値パターンの動的な想起が実現されている。

カオス四元数ニューロンモデルから構成されるカオス四元数連想メモリ¹⁸⁾は、自己想起型の連想メモリであり、ホップフィールドネットワーク²⁾と同じような構造をしている。ただし、各ニューロンはカオス四元数ニューロンモデルとなっており、内部状態、出力、重みなどが四元数を取れるようになっている。文献¹⁸⁾では、記憶させた複数のパターンを重ねた重畳パターンを入力として与えたときに、重畳パターンを構成する

複数のパターンを分離して想起できることが示されている。カオス四元数連想メモリは、カオス複素連想メモリ¹⁶⁾¹⁷⁾と同様に多値パターンの動的な想起が可能であるが、カオス四元数ニューロンモデルのパラメータに対して非常に敏感であり、動的な想起を実現するためのパラメータの設定が難しいため、文献¹⁸⁾では、動的な想起能力については触れられていない。

本研究では、カオス四元数連想メモリにおける動的な想起について調べ、その結果について報告する。

2 カオス四元数ニューロンモデル

ここでは、カオス四元数連想メモリで用いるカオス四元数ニューロンモデル¹⁸⁾について説明する。カオス四元数ニューロンモデルは、ニューロンの入出力や内部状態として四元数を扱えるようにカオスニューロンモデル¹⁴⁾を拡張したモデルである。

2.1 カオス四元数ニューロンモデル

カオス四元数ニューロンモデルのダイナミクスは

$$\mathbf{x}(t+1) = f \left(\mathbf{A}(t) - \alpha \sum_{d=0}^t k^d \mathbf{x}(t-d) - \boldsymbol{\theta} \right) \quad (1)$$

(ただし、 $\mathbf{A}(t), \mathbf{x}(t), \boldsymbol{\theta} \in \mathbb{H}$, $k, \alpha \in \mathbb{R}$)

のように表される。ここで、 $\mathbf{x}(t)$ は時刻 t におけるニューロンの出力、 $\mathbf{A}(t)$ は時刻 t における外部入力、 α は不応性に関するスケーリングファクタ、 k は時間減衰定数、 $\boldsymbol{\theta}$ はニューロンの閾値である。また、 \mathbb{H} は四元数の集合、 \mathbb{R} は実数の集合を表している。また、 $f(\cdot)$ は出力関数であり、

$$f(\mathbf{u}) = f^{(e)}(u^{(e)}) + f^{(i)}(u^{(i)})\mathbf{i} + f^{(j)}(u^{(j)})\mathbf{j} + f^{(k)}(u^{(k)})\mathbf{k} \quad (2)$$

$$\begin{aligned} f^{(e)}(u) &= f^{(i)}(u) = f^{(j)}(u) = f^{(k)}(u) \\ &= \tanh\left(\frac{u}{\varepsilon}\right) \end{aligned} \quad (3)$$

で与えられる。ここで、 ε は関数の傾きを決めるパラメータである。

2.2 時間的に変化する不応性のスケーリングファクタを有するカオス四元数ニューロンモデル

カオスニューロンモデル¹⁴⁾では、通常、不応性のスケーリングファクタは定数として扱われるが、文献¹⁵⁾

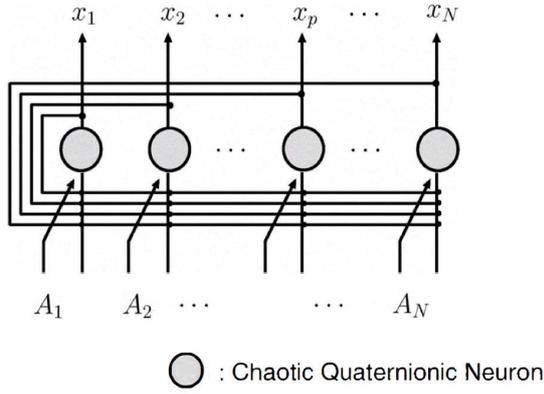


Fig. 1: カオス四元数連想メモリの構造

ではカオスニューロンモデルの不应性のスケールングファクタを時間的に変化させることで動的な想起能力が向上することが示されている。またカオス複素ニューロンモデル¹⁷⁾から構成されるカオス複素多方向連想メモリ^{?)}においても不应性のスケールングファクタを動的に変化させることで1対多の想起能力が向上することが知られている^{?)}。カオス四元数ニューロンモデルから構成されるネットワークにおいても、不应性のスケールングファクタを時間的に変化させることで動的な想起能力が向上する。

時間的に変化する不应性のスケールングファクタを有するカオス四元数ニューロンモデルのダイナミクスは、以下のように与えられる。

$$\mathbf{x}(t+1) = f \left(\mathbf{A}(t) - \alpha(t) \sum_{d=0}^t k^d \mathbf{x}(t-d) - \boldsymbol{\theta} \right) \quad (4)$$

(ただし、 $\mathbf{A}(t), \mathbf{x}(t), \boldsymbol{\theta} \in \mathbb{H}, k, \alpha(t) \in \mathbb{R}$)

ここで、 $\alpha(t)$ は時刻 t における不应性のスケールングファクタである。不应性のスケールングファクタの変化のさせ方としては様々なものが考えられるが、カオス四元数ニューロンモデルにおいても時間的に変化する不应性のスケールングファクタを有するカオス連想メモリ¹⁵⁾やカオス複素多方向連想メモリ^{?)}で用いられているのと同様に、

$$\alpha(t) = a + b \cdot \sin \left(c \cdot \frac{\pi}{12} \cdot t \right) \quad (5)$$

としている。ここで、 a, b, c は不应性のスケールングファクタの変化のさせ方を決めるパラメータである。

3 カオス四元数連想メモリ

ここでは、カオス四元数連想メモリ (Chaotic Quaternionic Associative Memory : CQAM)¹⁸⁾について説明する。

3.1 構造

カオス四元数連想メモリは、図1に示すように、ホップフィールドネットワークと同様の構造をしており、各ニューロンはカオス四元数ニューロンモデル¹⁸⁾となっている。

3.2 学習過程

カオス四元数連想メモリでは、重みを相関学習により決定する。ニューロン p と q の間の重み w_{pq} は

$$w_{pq} = \frac{1}{4N} \sum_{\mu=1}^P \mathbf{X}_p^{(\mu)} \otimes \mathbf{X}_q^{(\mu)*} \quad (6)$$

で与えられる。ここで、 N はニューロン数、 $\mathbf{X}_p^{(\mu)}$ は μ 番目のパターンの p 番目の成分、 $*$ は共役転置を表す。

3.3 想起過程

カオス四元数連想メモリにおいて、時刻 $t+1$ における p 番目のニューロンの出力は

$$\mathbf{x}(t+1) = f \left(\mathbf{v} \sum_{d=0}^t k_s^d \mathbf{A}_p(t-d) + \sum_{q=1}^N w_{pq} \sum_{d=0}^t k_m^d \mathbf{x}_q(t-d) - \alpha(t) \sum_{d=0}^t k_r^d \mathbf{x}_p(t-d) - \boldsymbol{\theta}_p \right) \quad (7)$$

で与えられる。ここで、 k_s, k_m, k_r は時間減衰定数、 $\alpha(t)$ は時刻 t における不应性のスケールングファクタ、 \mathbf{v} は外部入力からの重み、 \mathbf{A}_p は外部入力の p 番目の成分、 $\boldsymbol{\theta}_p$ はニューロンの p 番目の成分の閾値である。

4 計算機実験

4.1 重畳パターンの分離

ここでは、まず、カオス四元数連想メモリによる重畳パターンの分離の結果を示す。ここでは、図2のような5つのパターンを記憶させたモデルにおいて、birdとpandaの重畳パターン、whaleとpandaの重畳パターンを入力として与え、重畳パターンの分離を行った。図3を見ても分かるように、birdとpandaのパターン、whaleとpandaのパターンがそれぞれ想起されている。このモデルでは、ランダムパターンを用いて実験を行い、4つのパターンの重畳パターンの分離が行えることを確認している。

4.2 動的な想起

ここでは、複数のパターンをカオス四元数連想メモリに学習させ、それらのパターンのうちの一つを初期の入力として与え、外部入力としては何も与えない状態での動的想起能力を調べる実験を行った。

これまでのカオスニューロンモデルに基いたモデルにおいて想起能力を調べる実験では、1000時刻程度の時間の中で想起できるかどうかを調べていた。カオス四元数連想メモリでは、学習パターン数が2の場合には、両方のパターンを1000時刻程度の時間の中で想起できるようなカオス四元数ニューロンモデルのパラメータを試行錯誤により探すことは時間はかかるもの

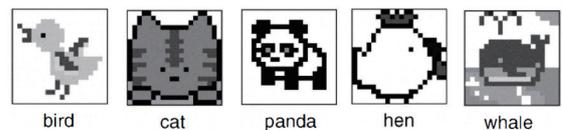
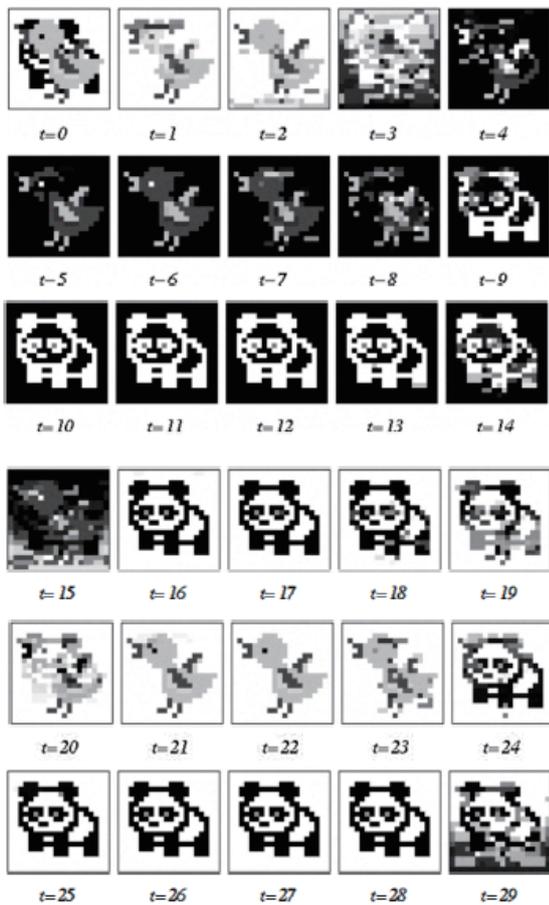


Fig. 2: 学習パターン

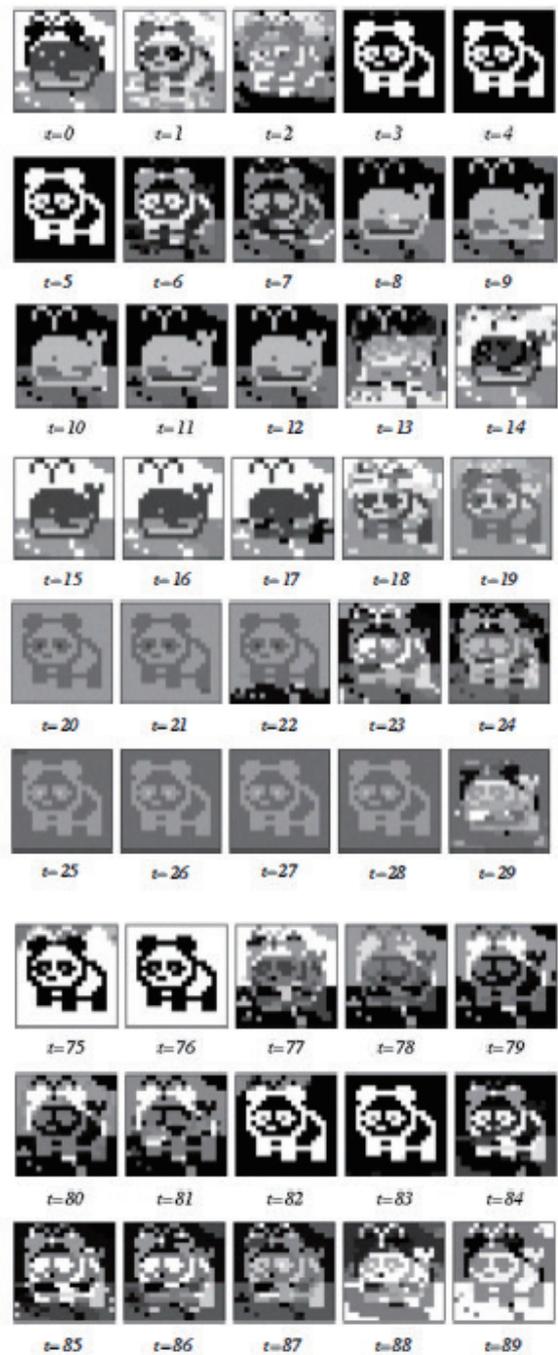


(a) *bird + panda*

の可能ではあった。しかしながら、学習パターンの数が3の場合ですら、1000時刻程度の時間の中ですべてのパターンを想起できるようなカオス四元数ニューロンモデルのパラメータを見つけることはできなかった。そこで、想起時間を長くとることで、想起率がどのように変化するかを調べる実験を行った。ここで、想起率とは、記憶したパターンに対する想起できたパターンの割合である。

図4はニューロン数が100のネットワークにランダムな16値のパターンを3つを記憶させたときの結果である。図を見ても分かるように、1000時刻まででは、想起率は0.57程度となっている。200000時刻あたりまでは想起率が徐々に高くなり、最終的には0.97程度、つまり、ほとんどの試行においてすべてのパターンが想起できるようになっている。

図5はニューロン数が100のネットワークにランダムな16値のパターンを4つを記憶させたときの結果である。図を見ても分かるように、1000時刻まででは、想起率は0.33程度となっている。これは、ほとんどの場合、入力されたパターンのみしか想起されず、まれにそれ以外のもう一つのパターンが想起されるといった状況である。200000時刻あたりまでは想起率が徐々に高くなってはいったものの、今回調べた範囲のパラメータでは、最終的にも想起率は0.60程度までしか到達しなかった。



(b) *whale + panda*

Fig. 3: 重畳パターンの分離

5 おわりに

本研究では、カオス四元数連想メモリにおける動的想起能力に関する検討を行った。これまでのカオスニューロンモデルに基いたモデルにおいて想起能力を調べる実験では、1000時刻程度の時間の中で想起できるかどうかを調べていたが、自己想起型の連想メモリでは、複数のパターンが想起されるまでにかかる時間が長いケースが多くみられ、特にカオス四元数連想メモリではその傾向が強いことが分った。

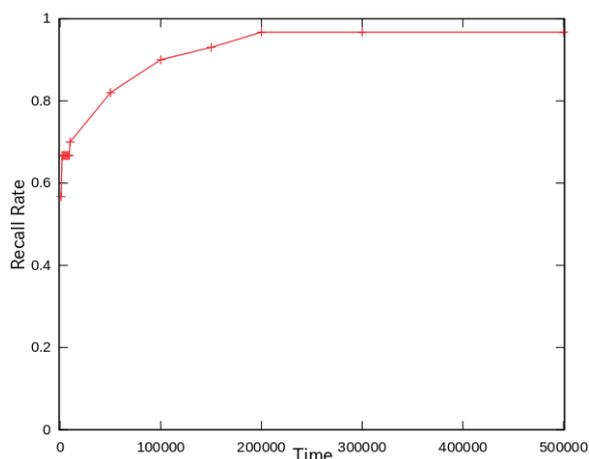


Fig. 4: 想起率の時間による変化 (パターン数 3)

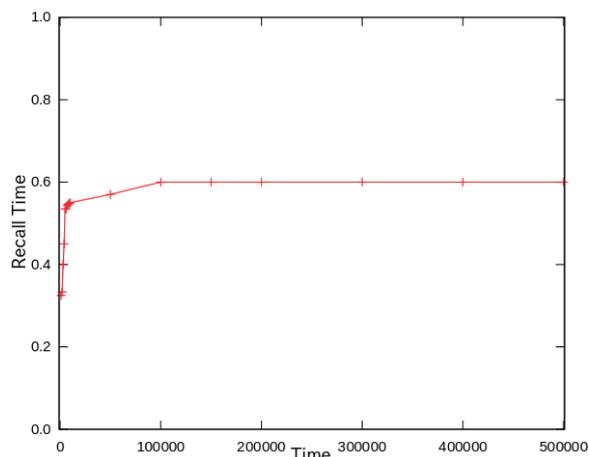


Fig. 5: 想起率の時間による変化 (パターン数 4)

参考文献

- 1) K. Nakano : “Associatron – a model of associative memory,” IEEE Transactions on Systems Man and Cybernetics, Vol.2, No.1, pp.380–388, 1972.
- 2) J. J. Hopfield : “Neural networks and physical systems with emergent collective computational abilities,” Proceedings of the National Academy of Sciences of the USA, Vol.79, pp.2554–2558, 1982.
- 3) B. Kosko : “Bidirectional associative memories,” IEEE Transactions on Systems Man and Cybernetics, Vol.18, No.1, pp.49–60, 1988.
- 4) M. Hagiwara : “Multidirectional associative memory,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Washington D.C., Vol.1, pp.3–6, 1990.
- 5) M. Hattori and M. Hagiwara : “Episodic associative memory,” Neurocomputing, Vol.12, pp.1–18, 1996.
- 6) 根本 幾, 河野 知志 : “神経回路網の複素数値化の試み,” 電子情報通信学会論文誌 D-II, Vol.J74, No.9, pp.1282–1288, 1991.
- 7) 橋本 直樹, 黒江 康明, 森 武宏 : “複素ニューラルネットワークに対するエネルギー関数について,” 電子情報通信学会技術報告, No.NC98-152, pp.121–128, 1999.
- 8) T. Isokawa, H. Nishimura, N. Kamiura and N. Matsui : “Fundamental properties of quaternionic Hopfield neural network,” International Journal of Neural Systems, Vol.18, No.2, pp.135–145, 2008.
- 9) 青木 宏之, 小林 幸夫 : “ペナルティ項を有する複素連想記憶モデルの性質,” 電子情報通信学会論文誌 A, Vol.J81, No.11, pp.1538–1546, 1998.
- 10) 青木 宏之 : “複素数体上に拡張したホップフィールド連想記憶の平衡状態の解析,” 電子情報通信学会論文誌 A, Vol.J78, No.9, pp.1238–1241, 1995.
- 11) S. Jankowski, A. Lozowski and J. M. Zurada : “Complex-valued multistate neural associative memory,” IEEE Transactions on Neural Networks, Vol.7, No.6, pp.1491–1496, 1996.
- 12) 黒江 康明 : “複素連想記憶のモデルとそのダイナミックス,” 電子情報通信学会学会誌, Vol.87, No.6, pp.454–459, 2004.
- 13) 合原 一幸編 : カオス –カオス理論の基礎と応用–, サイエンス社, 1990.
- 14) K. Aihara, T. Takabe and M. Toyoda : “Chaotic neural networks,” Physics Letter A, Vol.144, No.6 & 7, pp.333–340, 1990.
- 15) Y. Osana and M. Hagiwara : “Separation of superimposed pattern and many-to-many associations by chaotic neural networks,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Anchorage, Vol.1, pp.514–519, 1998.
- 16) M. Nakada and Y. Osana : “Chaotic complex-valued associative memory,” Proceedings of International Symposium on Nonlinear Theory and its Applications, Vancouver, 2007.
- 17) 中田正雄, 長名優子 : “カオス複素連想メモリによる多値パターンの動的想起の実現,” 知能と情報, Vol.20, No.3, pp.423–432, 2008.
- 18) Y. Osana : “Chaotic quaternionic associative memory,” Proceedings of IEEE and INNS International Joint Conference on Neural Networks, Brisbane, 2012.

リズム現象に対するパラメータ感度の解析法とその設計への応用

○森 禎弘 黒江康明 (京都工芸繊維大学)

Analysis Method of Parameter Sensitivities for Rhythmic Phenomena and Its Application to Their Design Problem

*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 and its application for rhythmic phenomena which are found in various systems such as physical systems, biological systems, 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. We have already proposed two analysis methods of period and phase sensitivities in rhythmic phenomena: one is based on the sensitivity equation and the other is based on the adjoint equation. In this paper we discuss an application of the sensitivity analysis methods. We have already proposed a design method of periodic phenomena with desired period and phase by using the sensitivity analysis method based on the sensitivity equation. In this paper, we propose a design method by using the sensitivity analysis method based on the adjoint equation. In order to show an efficiency of the proposed methods, we perform a numerical experiment in which we show two results obtained by using two proposed design methods. The results show that the proposed methods make it possible to obtain desired periodic phenomena accurately.

Key Words: Rhythmic phenomena, Period sensitivity, Phase sensitivity, Periodic orbit, Poincaré map, Optimal design

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見られる。またサーカディアン・リズムとよばれる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行われてきた。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。さらに生体だけでなく、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、あるいはこ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。また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で、すなわちパラメータ感度の解析は、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に基づく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、感度解析が必要となる。すなわち、評価関数のパラメータに関する微分値が必要となる。

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解析の研究はそれほど多くない。その理由は、システムの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であるためである。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、その現象の特徴的な量である周期と位相に対する感度解析法を二つ提案している^{4, 5)}。1つは感度方程式に基づく方法であり、他の一つは随伴方程式に基づく方法である。さらに、これらの方法に基づき感度を求めるための精度が良くかつ計算効率の高い感度解析アルゴリズムを提案している。

本稿の目的は、これまで筆者らが提案してきた感度解析法の応用について議論することである。具体的には、所望の周期および位相をもつリズム現象の設計法を議論する。この方法について、筆者らはすでに文献8)で感度方程式に基づく感度解析法を用いて設計する方法を提案している。本稿では、随伴方程式に基づく感度解析法を用いた設計法を提案する。提案法の有効性を示すため、提案した方法を用いて所望のリズム現象を設計した数値実験の結果を示す。特に文献8)の感度方程式に基づく感度解析法を用いた設計法と本稿で提案する随伴方程式に基づく感度解析法を用いた設計法の両方の方法を使って設計した数値実験結果を示し、それらの性能を比較、評価する。

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。

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

ここで、 $\mathbf{x}(t)$ は状態ベクトル、 \mathbf{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 ($\mathbf{x} \in \mathbb{R}^n$)、 p 次元ベクトル ($\mathbf{p} \in \mathbb{R}^p$) であるとする。 \mathbf{f} は非線形のベクトル関数である ($\mathbf{f}: \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$)。 $\mathbf{f}(\mathbf{x}, \mathbf{p})$ は、 \mathbf{x} および \mathbf{p} に関してそれぞれ 1 階連続微分可能とする。また、 $\phi(t, \mathbf{x}_0, \mathbf{p})$ を初期条件 $\mathbf{x}(0) = \mathbf{x}_0$ 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、 $\phi(t, \mathbf{x}_0, \mathbf{p})$ は次式を満たす。

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

(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 γ とし、この軌道上の一点を \mathbf{x}_γ とする。 γ が周期軌道であるということより、任意の点 $\mathbf{x}_\gamma \in \gamma$ に対し、

$$\mathbf{x}_\gamma = \phi(T, \mathbf{x}_\gamma, \mathbf{p}) \quad (3)$$

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 \mathbf{p} に依存して決まる。本稿では、(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 γ を対象として、その特徴的な量である周期と位相のパラメータ感度を求める方法およびその応用について議論する。

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^{9, 10)}。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 γ とする。 γ 上の一点を \mathbf{x}_{γ_0} とし、この点で γ と横断的に交わる超平面 (曲面) Σ を考える。 $U \subset \Sigma$ を点 \mathbf{x}_{γ_0} の近傍とする。点 $\mathbf{x} \in U$ に対して

$$\mathbf{P}(\mathbf{x}) = \phi(T, \mathbf{x}, \mathbf{p}) \quad (4)$$

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

明らかに、 $\mathbf{x}_{\gamma_0} = \mathbf{P}(\mathbf{x}_{\gamma_0})$ 、 $T(\mathbf{x}_{\gamma_0}) = T$ 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、 γ 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。

周期軌道に対し、上記の様に定義されるポアンカレ写像を導入することにより、周期軌道 γ の近傍での解の振る舞いの定性的な性質を調べることができる。

3 パラメータ感度の解析法

本節では、著者らが提案している周期軌道 γ の周期や位相に対するパラメータ感度を求める方法、すなわち感度解析法を説明する。これまで二つの方法、すなわち感度方程式に基づく方法⁶⁾と随伴方程式に基づく方法⁷⁾を提案している。周期軌道 γ の周期や位相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に対する依存性の表現を得る必要がある。そのため以下では、周期軌道の周期と位相に関するパラメータの依存性の数学的表現を導き、それを基にして感度を計算する方法を導く。

3.1 パラメータ感度の表現の導出

周期軌道 γ の位相に対するパラメータ依存性の表現の導出にあたっては、まず非線形システム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。まずこれを説明する。

(1) 式で表されるシステムの周期軌道 γ に対する位相を次のように定義する。 γ 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、適当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 \mathbf{p} に依存しているので、このことを明確にするため $t_{ev}(\mathbf{p})$ と表す。このように定義した事象 (event) が起こる時刻の位相 $\theta(\mathbf{p})$ を次のように定義する。

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

ここで、 $T(\mathbf{p})$ は周期軌道 γ の周期であり、これもパラメータ \mathbf{p} に依存して決まるので、このように表している。本稿では、このように定義した位相に関するパラメータ \mathbf{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 \mathbf{p} で微分すると感度の表現が次式のように得られる。

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

従って位相感度は、事象が起こる時刻に対する感度 $\partial t_{ev}(\mathbf{p}) / \partial \mathbf{p}$ および、周期に関する感度 $\partial T(\mathbf{p}) / \partial \mathbf{p}$ 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 \mathbf{x} とパラメータ \mathbf{p} に依存する $\mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}$ 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。

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

たとえば、状態 \mathbf{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 \mathbf{x} の i 番目の要素が極値をとるといような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、 $\mathcal{E}_v = \frac{d}{dt}x_i = f_i(\mathbf{x}, \mathbf{p})$ 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 $\partial t_{ev}(\mathbf{p})/\partial \mathbf{p}$ および、 $\partial T(\mathbf{p})/\partial \mathbf{p}$ の表現を導く。

周期感度 $\partial T(\mathbf{p})/\partial \mathbf{p}$ の表現は、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 γ 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。

$$\sigma(\mathbf{x}) = 0 \quad (8)$$

ここで、 σ は $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}$ なる写像である。対象とする周期軌道 γ に対し、ポアンカレ断面は適切に設定できるとし、 $\sigma(\mathbf{x})$ は既知であるとする。周期軌道とポアンカレ断面の交点 \mathbf{x}_{γ_0} は (3) 式を満たすので、次式が成立する。

$$\mathbf{x}_{\gamma_0} = \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \quad (9)$$

また \mathbf{x}_{γ_0} は Σ 上の点であることから

$$\sigma(\mathbf{x}_{\gamma_0}) = 0 \quad (10)$$

が成立する。(7) 式の $\mathbf{x}(t)$ は周期解なので

$$\mathbf{x}(t) = \phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p}) \quad (11)$$

と書け、このとき (7) 式は次の様を書くことができる。

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

ここで、パラメータ \mathbf{p} が $\Delta \mathbf{p}$ だけ変化し $\mathbf{p} + \Delta \mathbf{p}$ となったとする。このとき周期軌道 γ は、周期状態を維持したままポアンカレ断面 Σ との交点 \mathbf{x}_{γ_0} と周期 T 、および事象が起こる時刻 t_{ev} がそれぞれ

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

と変化したとする。すなわち次式が成立するとする。

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

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。

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

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

$$\sigma(\mathbf{x}_{\gamma_0} + \Delta \mathbf{x}_{\gamma_0}) = 0 \quad (15)$$

ここで、(13) 式の右辺をテーラー展開し (9) 式を用いることにより

$$\begin{aligned} \Delta \mathbf{x}_{\gamma_0} = & \frac{\partial}{\partial t} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta T + \frac{\partial}{\partial \mathbf{x}_0} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta \mathbf{x}_{\gamma_0} \\ & + \frac{\partial}{\partial \mathbf{p}} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta \mathbf{p} + O(\Delta^2) \end{aligned}$$

が得られる。ここで $O(\Delta^2)$ は、 Δ に関しての 2 次以上の高次の項を表す。この式を ΔT と $\Delta \mathbf{x}_{\gamma_0}$ を未知数とする方程式として整理すると次式が得られる。

$$\begin{aligned} \left(\mathbf{I} - \frac{\partial}{\partial \mathbf{x}_0} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \right) \Delta \mathbf{x}_{\gamma_0} - \frac{\partial}{\partial t} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta T \\ = \frac{\partial}{\partial \mathbf{p}} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta \mathbf{p} + O(\Delta^2) \end{aligned} \quad (16)$$

また (14) 式の左辺をテーラー展開し、(12) 式を用いることにより次式が得られる。

$$\begin{aligned} \frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial t} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta t_{ev} \\ + \frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial \mathbf{x}_0} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta \mathbf{x}_{\gamma_0} \\ + \frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) \Delta \mathbf{p} \\ + \frac{\partial}{\partial \mathbf{p}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \Delta \mathbf{p} + O(\Delta^2) = 0 \end{aligned} \quad (17)$$

なお、上式の左辺における $\mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p})$ はすべて $\mathcal{E}_v(\phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}), \mathbf{p})$ 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。さらに (15) 式の左辺をテーラー展開し、(10) 式を用いることにより、

$$\frac{\partial}{\partial \mathbf{x}} \sigma(\mathbf{x}_{\gamma_0}) \Delta \mathbf{x}_{\gamma_0} + O(\Delta^2) = 0 \quad (18)$$

が得られる。(16) 式と (17) 式および (18) 式を、 ΔT と $\Delta \mathbf{x}_{\gamma_0}$ と Δt_{ev} を未知数とした連立方程式と考えると、最終的に次ページの (19) 式が得られる。ここで記述の簡単のため、(19) 式の左辺の係数行列を \mathbf{S} 、右辺第 1 項の係数行列を \mathbf{V} として (19) 式を次のように表す。

$$\mathbf{S} \begin{bmatrix} \Delta \mathbf{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix} = \mathbf{V} \Delta \mathbf{p} + O(\Delta^2) \quad (20)$$

この方程式を解くことにより、次式が得られる。

$$\begin{bmatrix} \Delta \mathbf{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix} = \mathbf{S}^{-1} \mathbf{V} \Delta \mathbf{p} + O(\Delta^2) \quad (21)$$

これより感度が次の様に求まる。

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

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 \mathbf{S} および \mathbf{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ための感度方程式に基づくアルゴリズムおよび随伴方程式に基づくアルゴリズムを説明する。

$$\begin{aligned}
& \begin{bmatrix} \mathbf{I} - \frac{\partial}{\partial \mathbf{x}_0} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) & -\frac{\partial}{\partial t} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) & 0 \\ \frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial \mathbf{x}_0} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) & 0 & \frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial t} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) \\ \frac{\partial}{\partial \mathbf{x}} \sigma(\mathbf{x}_{\gamma_0}) & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}_{\gamma_0} \\ \Delta T \\ \Delta t_{ev} \end{bmatrix} \\
& = \begin{bmatrix} \frac{\partial}{\partial \mathbf{p}} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \\ -\frac{\partial}{\partial \mathbf{x}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p}) - \frac{\partial}{\partial \mathbf{p}} \mathcal{E}_v(\mathbf{x}(t_{ev}), \mathbf{p}) \\ 0 \end{bmatrix} \Delta \mathbf{p} + O(\Delta^2) \quad (19)
\end{aligned}$$

3.2 感度解析アルゴリズム

3.2.1 周期解の求解法

前項で導いた (21) 式の行列 \mathbf{S} および \mathbf{V} を求めるためには、まず周期軌道 γ を求める必要がある。これを求めるためには、通常、システムのモデルである微分方程式 (1) を、適当な初期条件を与えて解くことが行われる。実際、周期軌道の吸引領域 (アトラクター) に初期条件を選ぶことができると、(1) 式を定常状態まで解くことにより、周期解が得られる。ところがこの方法は、理論的には $t \rightarrow \infty$ の極限として周期解が得られるので、非常に時間がかかることになり計算効率が悪い。また、実際には周期状態に近づくに時間がかかるので、求まった周期解の精度が問題となり、この問題は感度の計算の精度に大きな影響を及ぼす。さらに、この方法は安定な周期解にしか適用できず、たとえばカオスアトラクターに埋め込まれた不安定な周期解の求解には用いることができない。以下では、これらの問題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 \mathbf{x}_{γ_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、 $\mathbf{y} = [\mathbf{x}'_{\gamma_0} \ T]'$ とおき

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

なる代数方程式を考える。ただし

$$\mathbf{F}(\mathbf{y}) = \begin{bmatrix} \mathbf{x}_{\gamma_0} - \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \\ \sigma(\mathbf{x}_{\gamma_0}) \end{bmatrix} \quad (22)$$

で、 $\mathbf{A}'(\mathbf{x}')$ は行列 \mathbf{A} (ベクトル \mathbf{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。

$$\mathbf{y}^{i+1} = \mathbf{y}^i - \mathbf{DF}(\mathbf{y}^i)^{-1} \mathbf{F}(\mathbf{y}^i) \quad (23)$$

ただし $\mathbf{DF}(\mathbf{y})$ は $\mathbf{F}(\mathbf{y})$ の \mathbf{y} に関するヤコビ行列で、次のように表される。

$$\mathbf{DF}(\mathbf{y}) = \begin{bmatrix} \mathbf{I} - \frac{\partial}{\partial \mathbf{x}_0} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) & -\frac{\partial}{\partial T} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p}) \\ \frac{\partial}{\partial \mathbf{x}} \sigma(\mathbf{x}_{\gamma_0}) & 0 \end{bmatrix} \quad (24)$$

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

$\mathbf{F}(\mathbf{y}^i)$ の第 1 項 $\phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ は、初期条件を $\mathbf{x}(0) = \mathbf{x}_{\gamma_0}^i$ とし (1) 式を $t = 0$ から $t = T^i$ まで解くことにより得られる。第 2 項 $\sigma(\mathbf{x}_{\gamma_0}^i)$ 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 $\mathbf{DF}(\mathbf{y}^i)$ の (1,1) ブロック要素の中の $\frac{\partial}{\partial \mathbf{x}_0} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ は、次のように計算できる。(2) 式の

両辺を \mathbf{x}_0 で微分すると、

$$\begin{aligned}
& \frac{\partial}{\partial \mathbf{x}_0} \phi(t, \mathbf{x}_0, \mathbf{p}) \\
& = \int_0^t \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\phi(\tau, \mathbf{x}_0, \mathbf{p}), \mathbf{p}) \frac{\partial}{\partial \mathbf{x}_0} \phi(\tau, \mathbf{x}_0, \mathbf{p}) d\tau \quad (25)
\end{aligned}$$

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

$$\frac{d}{dt} \mathbf{z}_{x_0}(t) = \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\phi(t, \mathbf{x}_0, \mathbf{p}), \mathbf{p}) \mathbf{z}_{x_0}(t), \quad \mathbf{z}_{x_0}(0) = \mathbf{I} \quad (26)$$

ここで、 \mathbf{I} は単位行列である。従って $\frac{\partial}{\partial \mathbf{x}_0} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ は、感度方程式と呼ばれる (26) 式の微分方程式を、 $t = 0$ から $t = T^i$ まで解いたときの解 $\mathbf{z}_{x_0}(T^i)$ として求まる。

一方、随伴方程式を導入して求める方法は、以下のようなになる⁵⁾。(26) 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。

$$\begin{aligned}
& \frac{d}{d\tau} \hat{\mathbf{z}}_{x_0}(\tau) = \mathbf{Df}' \cdot \hat{\mathbf{z}}_{x_0}(\tau), \quad \hat{\mathbf{z}}_{x_0}(0) = \mathbf{e}_j, \\
& j = 1, 2, \dots, n \quad (27)
\end{aligned}$$

ここで、 $\mathbf{Df} = \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p}), \mathbf{p})$, $\tau = T^i - t$ である。(27) 式を解いて得られる $\hat{\mathbf{z}}_{x_0}$ より、 $\frac{\partial}{\partial \mathbf{x}_0} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ は次式により求まる⁵⁾。

$$\{\mathbf{z}_{x_0 i}(T^i)\}_j = \{\hat{\mathbf{z}}_{x_0}(0)\}_i \quad (28)$$

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

またヤコビ行列 $\mathbf{DF}(\mathbf{y}^i)$ の (1,2) ブロック要素の $\frac{\partial}{\partial T} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ 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

$$\frac{\partial}{\partial T} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p}) = \mathbf{f}(\phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p}))$$

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 (23) 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 \mathbf{x}_{γ_0} と周期 T 、および周期軌道 $\phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p})$, $0 \leq t \leq T$ 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¹¹⁾。

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

行列 \mathbf{S} と \mathbf{V} の各要素も、対象とするシステムに対して初期状態 \mathbf{x}_0 に対する感度方程式およびパラメー

タ \mathbf{p} に対する感度方程式を導くことによりその計算法を導出することができる⁴⁾。また別の方法として、それらに対して随伴方程式を導くことによっても行列 \mathbf{S} と \mathbf{V} の各要素の計算法を導くことができる⁵⁾。ここでは、その一例として $\frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ を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めたポアンカレ断面と周期軌道の交点 \mathbf{x}_{γ_0} を初期状態として (1) 式を解くことにより周期軌道 $\phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p})$ 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} を用いて $\frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ は次のように求めることができる。 $\mathbf{z}_p(t) = \frac{\partial}{\partial \mathbf{p}} \phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p})$ とおくと、感度方程式

$$\begin{aligned} \frac{d}{dt} \mathbf{z}_p(t) &= \frac{\partial}{\partial \mathbf{x}} \mathbf{f}(\phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p}), \mathbf{p}) \mathbf{z}_p(t) \\ &+ \frac{\partial}{\partial \mathbf{p}} \mathbf{f}(\phi(t, \mathbf{x}_{\gamma_0}, \mathbf{p}), \mathbf{p}), \quad \mathbf{z}_p(0) = \mathbf{0} \end{aligned} \quad (29)$$

を導くことができ、これを $t=0$ から $t=t_{ev}$ まで解いたときの解 $\mathbf{z}_p(t_{ev})$ として、 $\frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ が求まる。

一方、随伴方程式を導入して求める場合は以下のようになる⁵⁾。新たな変数 $\hat{\mathbf{z}}_p$ を導入し、(29) 式の両辺と内積をとり、さらに、新たな変数 $\hat{\mathbf{u}}_p$ を導入して両辺に $\langle \mathbf{z}_{p_i}, \hat{\mathbf{u}}_p \rangle$ を加える。このとき $\frac{d}{dt} \langle \mathbf{z}_{p_i}, \hat{\mathbf{z}}_p \rangle = \langle \frac{d}{dt} \mathbf{z}_{p_i}, \hat{\mathbf{z}}_p \rangle + \langle \mathbf{z}_{p_i}, \frac{d}{dt} \hat{\mathbf{z}}_p \rangle$ が成り立つことと $\langle D\mathbf{f} \cdot \mathbf{z}_{p_i}, \hat{\mathbf{z}}_p \rangle = \langle \mathbf{z}_{p_i}, D\mathbf{f}' \cdot \hat{\mathbf{z}}_p \rangle$ が成り立つことに注意すると、最終的に次式が得られる。

$$\begin{aligned} &\langle \mathbf{z}_{p_i}, \frac{d}{dt} \hat{\mathbf{z}}_p + D\mathbf{f}' \cdot \hat{\mathbf{z}}_p + \hat{\mathbf{u}}_p \rangle \\ &= \langle \mathbf{z}_{p_i}, \hat{\mathbf{u}}_p \rangle - \langle \frac{\partial}{\partial p_i} \mathbf{f}, \hat{\mathbf{z}}_p \rangle + \frac{d}{dt} \langle \mathbf{z}_{p_i}, \hat{\mathbf{z}}_p \rangle \end{aligned} \quad (30)$$

ここで (30) 式の左辺に着目し

$$\frac{d}{dt} \hat{\mathbf{z}}_p = -D\mathbf{f}' \cdot \hat{\mathbf{z}}_p - \hat{\mathbf{u}}_p \quad (31)$$

とおき、さらに $\hat{\mathbf{z}}_p(T) = \mathbf{0}$ 、 $\hat{\mathbf{u}}_p(t) = \mathbf{e}_j \delta(T-t)$ とおき、 $\mathbf{z}_{p_i}(0) = \mathbf{0}$ 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。

$$\{\mathbf{z}_{p_i}(T)\}_j = \int_0^T \langle \frac{\partial}{\partial p_i} \mathbf{f}, \hat{\mathbf{z}}_p \rangle dt \quad (32)$$

(31) 式において $\tau = T-t$ とおき、入力にあたる部分 $\mathbf{e}_j \delta(0)$ は $\hat{\mathbf{z}}_p(\tau)$ 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、 $\frac{\partial}{\partial \mathbf{p}} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p})$ を求めるための随伴方程式は、以下の様になる。

$$\frac{d}{d\tau} \hat{\mathbf{z}}_p(\tau) = D\mathbf{f}' \cdot \hat{\mathbf{z}}_p(\tau), \quad \hat{\mathbf{z}}_p(0) = \mathbf{e}_j, \quad j = 1, 2, \dots, n \quad (33)$$

よって、随伴方程式は $\mathbf{z}_{p_i} (i = 1, 2, \dots, p)$ に対しすべて同じである。さらに (27) 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 $\mathbf{z}_{x_0}(\tau)$, $0 \leq \tau \leq T$ を用いて (32) 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 $\frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ を求めることができる。このときの随伴方程式は (33) 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 $\hat{\mathbf{z}}_p(0)$ を $\hat{\mathbf{z}}_p(0) = \mathbf{e}_j$ において随伴方程式 (33) を解き、その解 $\hat{\mathbf{z}}_p$ を用いて次式を計算することで求められる。

$$\{\mathbf{z}_{p_i}(t_{ev})\}_j = \int_0^{t_{ev}} \langle \frac{\partial}{\partial p_i} \mathbf{f}, \hat{\mathbf{z}}_p \rangle dt \quad (34)$$

なお、先述の方法で周期軌道が求められていると、ニュートン法において $T^i \rightarrow T$, $\mathbf{x}_{\gamma_0}^i \rightarrow \mathbf{x}_{\gamma_0}$ となるので、ニュートン法が収束した時点で行列 \mathbf{S} , \mathbf{V} の要素に出てくるもののうち、 $\frac{\partial}{\partial \mathbf{x}_0} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p})$, $\frac{\partial}{\partial t} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p})$, $\frac{\partial}{\partial \mathbf{x}} \sigma(\mathbf{x}_{\gamma_0})$ が得られている。

4 感度解析法の応用

4.1 リズム現象の設計への応用

本項では、提案した感度解析法の応用としてリズム現象の設計問題をとりあげる。ここで、とりあげる問題は、所望の周期と位相をもつリズムを実現する問題で、次のように定式化する。設計したいリズムの所望の周期と位相をそれぞれ、 T_d と θ_d とする。このとき、次のような評価関数を設定する。

$$J = \frac{1}{2} \alpha (T - T_d)^2 + \frac{1}{2} \beta (\theta - \theta_d)^2 \quad (35)$$

ここで、 $\alpha > 0$ と $\beta > 0$ は適当な重み係数である。このように評価関数を設定すると所望のリズムを実現する問題は、この評価関数を最少とするパラメータ \mathbf{p} を求める次のような最適化問題となる。

$$\min_{\mathbf{p}} J \quad (36)$$

この最適化問題を解く方法は、これまで多くの方法が提案されているが、最も一般的な方法は勾配法に基づく方法で、最急降下法、共役勾配法、準ニュートン法がその代表的な方法である。これらの方法を適用して最適パラメータを求めるためには、目的関数のパラメータに関する勾配を如何に求めるかが問題で、この勾配を求めるために周期と位相のパラメータの感度を求めることが必要になる。これらの計算に、前節までに説明した感度解析法を用いることができる。特に、随伴方程式に基づく感度解析法を用いた設計法は、次のようなアルゴリズムとしてまとめられる。

- Step 1. 適切な初期値 T^0 , $\mathbf{x}_{\gamma_0}^0$, \mathbf{p} を設定する。 $i = 0$ とする。
- Step 2. $\mathbf{F}(\mathbf{y}^i)$ と $D\mathbf{F}(\mathbf{y}^i)$ を求める。このとき、 $\frac{\partial}{\partial \mathbf{x}_0} \phi(T^i, \mathbf{x}_{\gamma_0}^i, \mathbf{p})$ は、随伴方程式 (27) を $\tau = 0$ から $\tau = T^i$ まで解き、(28) 式により求める。
- Step 3. $D\mathbf{F}(\mathbf{y}^i)^{-1} \mathbf{F}(\mathbf{y}^i)$ のノルムが十分小さければ、 $T = T^i$, $\mathbf{x}_{\gamma_0} = \mathbf{x}_{\gamma_0}^i$ として Step 5. に進む。
- Step 4. (23) 式により \mathbf{y}^i を更新し、 $i = i + 1$ として Step 2. に戻る。

Step 5. 得られた T と θ より, 目的関数 J の値を求める.

Step 6. 行列 \mathbf{S} と \mathbf{V} を求める. ただし, $\frac{\partial}{\partial \mathbf{p}} \phi(T, \mathbf{x}_{\gamma_0}, \mathbf{p})$ は, Step 2. で求めた $\hat{\mathbf{x}}_{x_0}$ を用いて, (32) 式よりを求める. また, $\frac{\partial}{\partial \mathbf{p}} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ と $\frac{\partial}{\partial \mathbf{x}_0} \phi(t_{ev}, \mathbf{x}_{\gamma_0}, \mathbf{p})$ は, $\tau = t_{ev} - t$ として随伴方程式 (28) 式を解き, (34) 式より求める.

Step 7. $\mathbf{S}^{-1} \mathbf{V}$ を計算して, 感度を求める.

Step 8. 適当な勾配法に基づく方法でパラメータ \mathbf{p} を更新する.

Step 9. \mathbf{p} の更新量が十分小さければ終了し, そうでなければ Step 2. に戻る.

4.2 数値実験例

本項では, 提案法の有効性を示すために, 次式で表される Brusselator を対象とした数値実験例を示す.

$$\begin{cases} \frac{d}{dt} x_1(t) = p_1 - (p_2 + 1)x_1(t) + x_1^2(t)x_2(t) \\ \frac{d}{dt} x_2(t) = p_2 x_1(t) - x_1^2(t)x_2(t) \end{cases} \quad (37)$$

ここでのパラメータは $\mathbf{p} = [p_1 \ p_2]'$ で, $p_1 > 0, p_2 > 0$ である. このシステムは, $[p_1 \ \frac{p_2}{p_1}]'$ を平衡点としてもつ. この平衡点は $p_2 > p_1^2 + 1$ のとき不安定で, このとき安定なリミットサイクルが存在する¹²⁾. このリミットサイクルが設計対象のリズム現象とする.

ここでは, リミットサイクルにおいて $x_1 = 3.0$ となる二つの状態のうち, x_2 の値が小さい方の状態を位相を測る基準, すなわちこの点の位相の値を 0 とする. また, 位相を求める事象として, 位相 0 の点から始まる軌道が初めて $x_2(t) = 3.0$ となる事象とし, すなわち $\mathcal{E}_v(\mathbf{x}, \mathbf{p}) = x_2(t) - 3.0$ とし, その位相を θ_{ev} とする. 所望の周期 T_d と位相 θ_d はそれぞれ $T_d = 5.09$ と $\theta_d = 0.960$ と与えられるとする.

リミットサイクルとポアンカレ断面との交点の位相が 0 となるように, ポアンカレ断面 Σ を $\sigma(\mathbf{x}) = x_1 - 3.0 = 0$ とし, 式 (36) の最適化問題を勾配法に基づく方法の一つである準ニュートン法を用いて解くことで設計する. 準ニュートン法としては, DFP 法を用いる. パラメータ \mathbf{p} を $\mathbf{p} = [1.0 \ 3.0]'$ とすると, Fig. 1 と Fig. 2 に示すような安定なリミットサイクルをもち, その周期と位相は 7.156919932 と 2.022974683 である. このパラメータを初期値に, 目的関数の重み α と β を $\alpha = \beta = 1.0$ と設定し, 感度方程式に基づく方法および随伴方程式に基づく方法のそれぞれで最適化問題 (36) を解いた. 感度方程式に基づく方法で解いた場合, 11 回の更新により, 評価関数の値は $3.5949346453 \times 10^{-17}$ と十分小さな値となり, システムのパラメータとして $[p_1 \ p_2]' = [2.0027748631 \ 6.0052294588]'$ が得られた. 随伴方程式に基づく方法で解いた場合, 同じく 11 回の更新により, 評価関数の値は $3.5462311292 \times 10^{-17}$ と十分小さな値となり, システムのパラメータとして $[p_1 \ p_2]' = [2.0027748632 \ 6.0052294590]'$ が得られた. 感度方程式を用いた場合と随伴方程式を用いた場合の目的関数値とその時に得られている目的関数の勾配の値をそれぞれ Table 1 と Table 2 に示す. また, 随伴

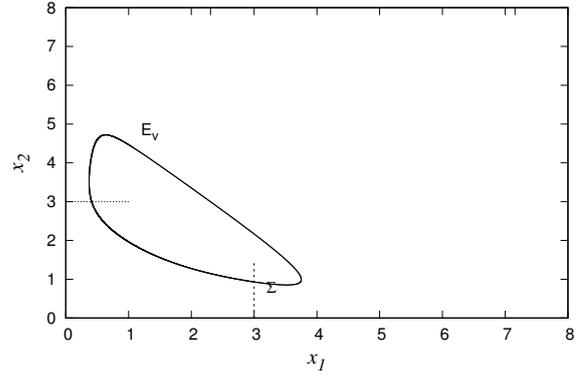


Fig. 1: Phase space plot of stable limit cycle of Brusselator (37) for $[p_1 \ p_2]' = [1.0 \ 3.0]'$.

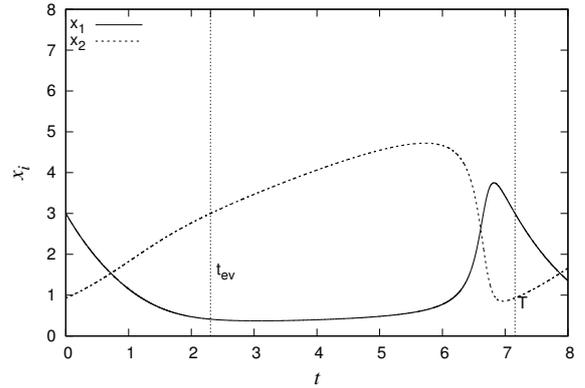


Fig. 2: Time evolutions of stable limit cycle of Brusselator (37) for $[p_1 \ p_2]' = [1.0 \ 3.0]'$.

方程式に基づく方法を用いて解いた場合の目的関数の収束の様子を Fig. 3 に示す. Table 1 と Table 2 から, 二つの提案法で得られた値は誤差の範囲で一致しており, 両者は同等の数値精度を実現できていることがわかる. 得られたパラメータ値におけるシステムの安定なリミットサイクルを Fig. 4 と Fig. 5 に示す. 得られたリミットサイクルの周期と位相を 3.2.1 節の周期解の求解法により求めると, 感度方程式に基づく設計法で得られたパラメータでは $T = 5.0900001561$ と $\theta = 0.95999998625$ であった. 一方, 随伴方程式に基づ

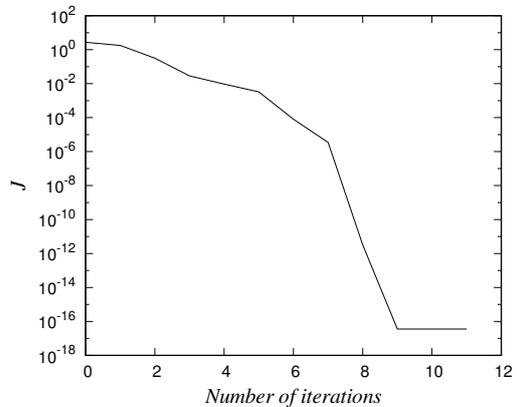


Fig. 3: An example of convergence behavior of objective function J obtained by using the sensitivity analysis method based on the adjoint equation.

Table 1: Variations of values of objective function J and gradient $\frac{\partial J}{\partial p}$ obtained by using the sensitivity analysis method based on the sensitivity equation.

Number of iterations	J	$\frac{\partial J}{\partial p_1}$	$\frac{\partial J}{\partial p_2}$
0	2.7010362990	-17.209306407	1.0858626920
1	1.7599423560	-3.3457746772	-1.8415342379
2	0.31798948968	-3.7749718547	0.83203640454
3	0.028614599676	-0.018053405956	-0.081915694556
4	0.0094230862518	-0.38721710478	0.094198573611
5	0.0032816869001	-0.0056444836795	-0.023205917906
6	$8.0228197320 \times 10^{-5}$	-0.043728854069	0.013454613876
7	$3.4847968305 \times 10^{-6}$	$-2.0996190826 \times 10^{-4}$	$-6.8237689139 \times 10^{-4}$
8	$3.2489097129 \times 10^{-12}$	$-9.1527853379 \times 10^{-6}$	$2.9732398038 \times 10^{-6}$
9	$3.5949327637 \times 10^{-17}$	$-6.8820156094 \times 10^{-10}$	$-2.1808059753 \times 10^{-9}$
10	$3.5949346453 \times 10^{-17}$	$-6.8819807580 \times 10^{-10}$	$-2.1808077372 \times 10^{-9}$
11	$3.5949346453 \times 10^{-17}$	$-6.8819807580 \times 10^{-10}$	$-2.1808077372 \times 10^{-9}$

Table 2: Variations of values of objective function J and gradient $\frac{\partial J}{\partial p}$ obtained by using the synthesis method based on the adjoint equation.

Number of iterations	J	$\frac{\partial J}{\partial p_1}$	$\frac{\partial J}{\partial p_2}$
0	2.7010362990	-17.209303399	1.0858632582
1	1.7599424184	-3.3457771630	-1.8415334743
2	0.31798886619	-3.7749611874	0.83203516975
3	0.028614197263	-0.018054503232	-0.081914594278
4	0.0094228515106	-0.38721166058	0.094197550300
5	0.0032815900972	-0.0056447722528	-0.02.3205412915
6	$8.0205670841 \times 10^{-5}$	-0.043722921203	0.013452880763
7	$3.4831885650 \times 10^{-6}$	$-2.0991085588 \times 10^{-4}$	$-6.8221928550 \times 10^{-4}$
8	$3.2471331810 \times 10^{-12}$	$-9.1502689613 \times 10^{-6}$	$2.9724407671 \times 10^{-6}$
9	$3.5462322333 \times 10^{-17}$	$-6.9935946258 \times 10^{-10}$	$-2.1608346040 \times 10^{-9}$
10	$3.5462311292 \times 10^{-17}$	$-6.9936152122 \times 10^{-10}$	$-2.1608335629 \times 10^{-9}$
11	$3.5462311292 \times 10^{-17}$	$-6.9936152122 \times 10^{-10}$	$-2.1608335629 \times 10^{-9}$

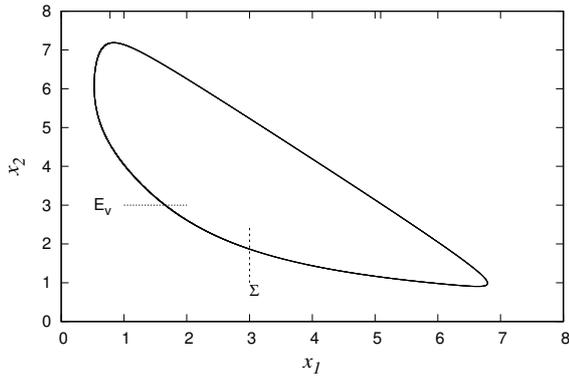


Fig. 4: Phase space plot of stable limit cycle of Brusselator (37) for $[p_1 p_2]' = [2.0027748632 6.0052294590]'$.

く設計法で得られたパラメータでは $T = 5.0900001561$ と $\theta = 0.95999998619$ と求まり、所望の周期と位相をもつリズム現象を設計できていることがわかる。

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

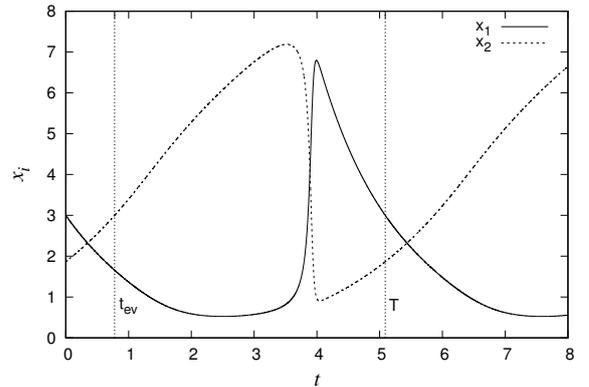


Fig. 5: Time evolutions of stable limit cycle of Brusselator (37) for $[p_1 p_2]' = [2.0027748632 6.0052294590]'$.

の問題の解決を図ろうとする研究が盛んになっている。その際、重要なツールの一つが感度解析法である。筆者らはこれまで、リズム現象の感度解析法として、特にリズムの周期と位相のパラメータに関する感度を求める方法を提案している。本稿では、その一つの応用として、所望のリズム現象を設計する方法について議論した。すなわち、所望の周期と位相を持つリズム現象を感度解析法を利用して実現する方法を提案した。

本稿の設計問題に対して、筆者らはすでに文献8)で感度方程式に基づく感度解析法を用いて設計する方法を提案している。本稿では、随伴方程式に基づく感度解析法を用いた設計法を提案した。さらに、これらの両方の方法を使って設計した数値実験結果を示した。その結果から、これらが同等の数値精度を実現できていることがわかった。

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) 森, 黒江: リズム現象における位相・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講演論文集, 60/67 (2015)
- 8) 黒江, 森: リズム現象に対するパラメータ感度の解析法とその応用, 第26回インテリジェント・システム・シンポジウム講演論文集, 265/279 (2016)
- 9) J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Field*. New York: Springer-Verlag (1983)
- 10) 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)
- 11) 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)
- 12) 郡, 森田: 生物リズムと力学系, 共立出版 (2011)

四元数ニューラルネットを用いた筋電位による前腕姿勢の推定

○小川 毅彦 ハフィズディン フィルダウス (拓殖大学)

Estimation of Forearm Posture by EMG Using a Quaternion Neural Network

* T. Ogawa and H. Firdaus (Takushoku University)

Abstract — Quaternions are often used for the posture expression of an object in the three dimensional space in the robot control and the computer graphics. Quaternion neural networks can learn a relation of the input/output relation which is expanded to quaternions. In this study, we introduced a quaternion neural network to the forearm posture estimation by the EMG signal and confirmed the effect by a simulation.

Key Words: Quaternion neural network, forearm posture, EMG.

1 はじめに

ロボット制御やコンピュータグラフィックスなど3次元空間内でのモデルの姿勢表現にしばしば四元数を用いられる。四元数を用いることで、回転行列を用いる際に生じる問題が解決できる¹⁻³⁾。また、ニューラルネットの四元数への拡張が研究されている^{4,7)}。四元数ニューラルネットを用いることで、四元数入出力間の非線形関係を学習・推定できる。

生体動作の解析や各種機器とのインタフェースとして筋電位が用いられる。特に皮膚表面上に貼付した表面電極で計測する表面筋電位は、筋肉を制御する信号を非侵襲的に直接計測することができるため、さまざまな応用が研究されている^{8,9)}。人間の前腕は、各種関節の回転によって3次元空間内でさまざまな姿勢をとることができる¹⁰⁾。そこで、四元数を用いることで効果的に前腕の姿勢を表現することができる。

本研究では、表面筋電位による前腕姿勢の推定を考える。これにより、表面筋電位による姿勢の評価や、さらにコンピュータとのインタフェースや、ロボットの遠隔制御などの応用が考えられる。表面筋電位と姿勢との関係は一般に非線形であり、ニューラルネットによる学習・推定が研究されている¹¹⁻¹³⁾。本研究では、表面筋電位からの前腕姿勢の推定の問題に、四元数ニューラルネットを適用することを考え、シミュレーションによって効果を示す。

2 四元数とニューラルネット

四元数は高次複素数であり、1つの実部と3つの虚部からなる。3つの虚数単位を i, j, k とすると、ある四元数を $x = x_0 + ix_1 + jx_2 + kx_3$ と書くことができる。ここで x_0, x_1, x_2, x_3 は四元数の各成分を表す実数である。虚数単位は、 $i^2 = j^2 = k^2 = ijk = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j$ を満たす。四元数は乗法に関して結合則を満たすが交換則を満たさないという特徴がある。

2.1 四元数とモデルの回転

3次元空間内のモデルの姿勢表現には、行列や四元数を用いられる。行列を用いて姿勢を表現する場合、回転を表現する方法として回転行列がある。回転行列

では、1つの角変位を互いに垂直な3本の軸の周りの3つの一連の回転として定義する。例えば、回転軸を a 、回転角を θ とし、回転行列を $R(\theta, a)$ とし、 x 軸周りに回転角 α 、 y 軸周りに回転角 β 、 z 軸周りに回転角 γ の回転をそれぞれ $R(\alpha, x)$ 、 $R(\beta, y)$ 、 $R(\gamma, z)$ とすると、対象のモデルを x 軸、 y 軸、 z 軸の順に回転させる回転行列は

$$R(\gamma, z)R(\beta, y)R(\alpha, x) = \begin{pmatrix} C_\beta C_\gamma & -C_\alpha S_\gamma + S_\alpha S_\beta C_\gamma & S_\alpha S_\gamma + C_\alpha S_\beta C_\gamma \\ C_\beta S_\gamma & C_\alpha C_\gamma + S_\alpha S_\beta S_\gamma & -S_\alpha C_\gamma + C_\alpha S_\beta S_\gamma \\ -S_\beta & S_\alpha C_\beta & C_\alpha C_\beta \end{pmatrix} \quad (1)$$

となる。ここで $S_\theta = \sin\theta$ 、 $C_\theta = \cos\theta$ とする。 y 軸の回転角が $\beta = \pi/2$ のとき、

$$R(\gamma, z)R\left(\frac{\pi}{2}, y\right)R(\alpha, x) = \begin{pmatrix} 0 & S_{\alpha-\gamma} & C_{\alpha-\gamma} \\ 0 & C_{\alpha-\gamma} & -S_{\alpha-\gamma} \\ -1 & 0 & 0 \end{pmatrix} \quad (2)$$

となる。対象のモデルを $P = (x, y, z)^T$ とすると、

$$P' = \begin{pmatrix} S_{\alpha-\gamma} \cdot y + C_{\alpha-\gamma} \cdot z \\ C_{\alpha-\gamma} \cdot y - S_{\alpha-\gamma} \cdot z \\ -x \end{pmatrix} \quad (3)$$

となり、 $z = -x$ の平面上での回転となる。すなわち、2つの回転軸が一致するジンバルロック状態となる。

一方、四元数を用いる場合は、単位ベクトルとして表現される回転軸を中心とする回転として定義する。回転軸を $a = ia_x + ja_y + ka_z$ 、回転角を θ とすると、回転を表す四元数は $q = \cos\theta/2 + a\sin\theta/2$ となる。 x 軸周りに回転角 α 、 y 軸周りに回転角 β 、 z 軸周りに回転角 γ の回転を表す四元数 $q_{x,\alpha}$ 、 $q_{y,\beta}$ 、 $q_{z,\gamma}$ で、対象とするモデル p を x 軸、 y 軸、 z 軸の順に回転させる場合、

$$p' = q'_{z,\gamma}(q'_{y,\beta}(q'_{x,\alpha}p)q_{x,\alpha})q_{y,\beta}q_{z,\gamma} = (q'_{z,\gamma}q'_{y,\beta}q'_{x,\alpha})p(q_{x,\alpha}q_{y,\beta}q_{z,\gamma}) \quad (4)$$

となる。ここで $q_{x,\alpha} = \cos\alpha/2 + isin\alpha/2$ 、 $q_{y,\beta} = \cos\beta/2 + jsin\beta/2$ 、 $q_{z,\gamma} = \cos\gamma/2 + ksin\gamma/2$ より、

$$q_{x,\alpha}q_{y,\beta}q_{z,\gamma} = (C_\alpha C_\beta C_\gamma - S_\alpha S_\beta S_\gamma) + i(C_\alpha S_\beta S_\gamma - S_\alpha C_\beta C_\gamma) + j(C_\alpha S_\beta C_\gamma - S_\alpha C_\beta S_\gamma) + k(C_\alpha C_\beta S_\gamma - S_\alpha S_\beta C_\gamma) \quad (5)$$

と表せる. ここでは $S_\theta = \sin\theta/2$, $C_\theta = \cos\theta/2$ とする.
 y 軸の回転角が $\beta = \pi/2$ のとき,

$$q_{x,\alpha} q_{y,\frac{\pi}{2}} q_{z,\gamma} = \frac{1}{\sqrt{2}} (C_{\alpha+\gamma} + iS_{\alpha+\gamma} + jC_{\alpha+\gamma} + kS_{\alpha+\gamma}) \quad (6)$$

となり, 原点を中心とした立体中の値を取る. すなわちジンバルロックが回避される.

四元数を用いる場合のもう1つの特徴的な演算として球面線形補間がある. 回転軸 a , 回転角 α の回転を $R(\alpha, a)$, 回転軸 b , 回転角 β の回転を $R(\beta, b)$ とすると, これらを補間する回転は,

$$R(\gamma, c) = R((1-\tau)\beta, b) \cdot R(\tau\alpha, a) \quad (7)$$

となる. 球面線形補間は線形補間と異なり, 2つの回転間を球面上の線形補間として滑らかに表現できる.

2.2 四元数ニューラルネット

さまざまな高次元ニューラルネットモデルが提案されている^{4,6)}. 本研究では, 入出力四元数間の写像関係の学習を考えるため, 層状の四元数ニューラルネットを考える.

四元数ニューロンの伝達関数として, 四元数の各部にシグモイド関数を適用する. すなわち, 入力 $s = s_0 + is_1 + js_2 + ks_3$ に対して

$$g(s) = f(s_0) + if(s_1) + jf(s_2) + kf(s_3), \quad (8)$$

ただしシグモイド関数は逆正接関数

$$f(u) = \frac{1-e^{-u}}{1+e^{-u}} \quad (9)$$

とし, これを中間・出力層に用いる. 学習時は, 四元数に拡張した誤差逆伝搬法により荷重の更新を行う. 誤差関数は四元数出力の2乗誤差

$$E = \frac{1}{2} \sum_r (d_r - y_r)^2 \quad (10)$$

とする. ここで d_r と y_r は, それぞれ r 番目素子に与えられた出力とネットワーク出力を表す. さらに中間・出力層間荷重の修正量 Δw_{qr} と入力・中間層間荷重の修正量 Δw_{pq} はそれぞれ,

$$\begin{aligned} \Delta w_{qr} &= ((d_r - y_r) \cdot (1 - y_r) \cdot (1 + y_r)) \cdot \bar{v}_q \\ \Delta w_{pq} &= ((1 - v_q) \cdot (1 + v_q) \cdot \sum_r \delta_r \cdot \bar{w}_{qr}) \cdot \bar{x}_p \end{aligned} \quad (11)$$

とする. ただし,

$$\delta_r = (d_r - y_r) \cdot (1 - y_r) \cdot (1 + y_r) \quad (12)$$

とする. ここでは, “ \cdot ”の記号で四元数の各要素の積を表す. また, v_q および x_p は, それぞれ q 番目中間素子からの出力および, p 番目入力を表す. $\mathbf{1}$ は各要素が1の四元数とする. これらの荷重修正量に基づいて,

$$w_{new} = w_{old} - \varepsilon \Delta w \quad (13)$$

に基づき入力を更新する. ε は学習係数を意味する. この手順を繰り返すことで, 四元数に拡張された層状ニューラルネットにより入出力関係を学習できる.

3 筋電位と前腕姿勢

筋肉は生体動作を生成するアクチュエータであり, 筋電位は筋肉を制御する神経パルスの重畳である⁹⁾. そこで, 筋電位を計測・解析することによって, 筋肉の制御意図を推定することができる.

筋肉の制御信号である筋電位と, 筋肉の収縮の関係については, 筋の全長に変化がない等尺性収縮と, 筋張力が変化せずに収縮する等張性収縮, 筋張力が変化しても関節運動が一定速度である等速性収縮の3つに分けられる¹⁰⁾. 本研究では, 張力と変位の関係をモデル化したいので, 単純に等張性収縮を想定する.

筋肉の収縮においては, 神経パルスの頻度に依存して張力が発生するため, 張力は神経パルスの関数と考えることができる. また, 変位は張力の関数として表現できる. そのため, 変位はパルスの関数であると言える. 言い換えると, 体の部位の変位は, 筋電位の関数として表現することができる. 本研究では, 筋電位に基づいて前腕姿勢すなわち腕の先端の変位を推定することを考える.

3.1 前腕姿勢と筋肉

人間の前腕は, 各種関節の回転により, 3次元空間での様々な姿勢をとることが可能である. また, 前腕の姿勢や動作によって, 様々なコミュニケーションや制御インタフェースとしての応用が考えられる. 従って, その前腕動作を生成する筋電位と前腕姿勢の関係を学習・推定することは意義があるものとする.

本研究では, 前腕の3次元空間内での姿勢と筋電位の関係を学習・推定することを考える. 具体的には, 前腕のロール・ピッチ・ヨー回転を対象とし, 筋肉としては上腕二頭筋 (biceps brachii), 円回内筋 (pronator teres), 棘下筋 (infraspinatus) を考える¹⁴⁾. これらの筋肉はそれぞれの回転に対して支配的に働くが, 実際には複数の筋肉の協調によって回転動作が発生する. 従って, 表面電極によって筋電位を測定する場合, 実際には複数の筋肉からの重畳波形となる. Fig. 1 に体軸と前腕の回転を示す.

本研究では, 上腕二頭筋, 円回内筋, 棘下筋の筋電位をそれぞれ e_1, e_2, e_3 とし, それに伴う回転をそれぞれ四元数で表し, q_x, q_y, q_z とする. これらの筋電位から前腕の回転を学習・推定することを考える. ただし, 今回は四元数ニューラルネットでの回転姿勢の学習・推定の動作を確認することを目的とし, 実波形でなく疑似的な筋電位でシミュレーションを行う.

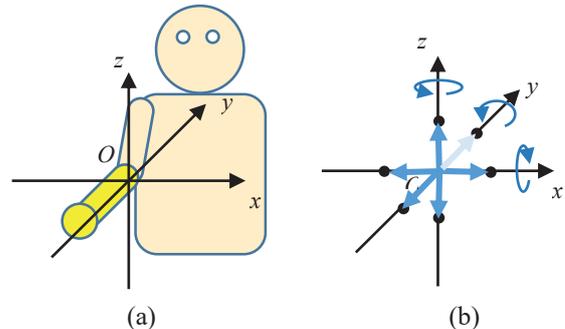


Fig 1. Body axes and forearm's rotation (a) body axes, and (b) forearm's rotation around each axis.

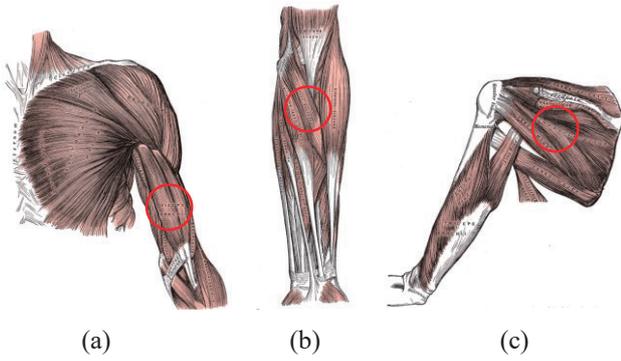


Fig 2. Corresponding muscles (a) biceps brachii, (b) pronator teres, and (c) infraspinatus ¹⁵.

4 姿勢推定のための四元数ニューラルネット

本研究では、筋肉への制御信号である筋電位と前腕姿勢の関係性を四元数ニューラルネットによって学習・推定することを目的とする。前腕の姿勢を、3つの筋肉によって制御される関節の回転によって表現することを考え、3チャンネル筋電位と関節の回転を四元数で表現する。そのため、これらの間の関係を四元数ニューラルネットで表現する。

筋活動については等張性収縮を想定し、変位は筋電位に比例するものとする。今回は筋電位からのニューラルネットによる姿勢推定の基本的な動作を確認するために、筋電位の実波形ではなく疑似電位を考え、単純化のため振幅の値を単に ± 1 と仮定する。これは、等張性収縮で得た筋電位を全波整流後にローパスフィルタを通して処理することを想定している。

本研究では、筋肉への制御信号である筋電位と関節の回転変位の間の関係をモデル化する。まず、ある筋電位 e_i と空間座標 y_i を考え、この関係を

$$\dot{x}_i = Ax_i + Be_i, y_i = Cx_i \quad (14)$$

と記述する。この関係は入力を e_i と x_i 、出力を y_i とするリカレント型ニューラルネットでモデル化できる。本研究では、3チャンネル筋電位と空間座標をそれぞれ $e = ie_1 + je_2 + ke_3$, $y = iy_1 + jy_2 + ky_3$ のように四元数として取り扱い、Fig.3 に示す四元数に拡張したりカレント四元数ニューラルネットでモデル化する。

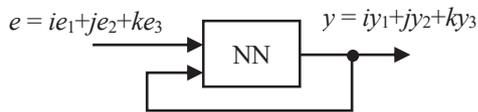


Fig.3 Recurrent quaternion neural network that models the relations between EMG signals and displacements.

5 シミュレーション

四元数ニューラルネットによる筋電位からの姿勢推定の動作を示すために、計算機シミュレーションを行った。ネットワークは四元数の2入力1出力とし、出力が再び入力される。ネットワーク構成を Table 1 に示す。学習は、3チャンネル筋電位に相当する四元数

$e = ie_1 + je_2 + ke_3$ と、各軸上 ± 1 の点を表す四元数 $x = ix_1 + jx_2 + kx_3$ の入力に対し、この点を各軸周り $\pi/2$ 回転した点を表す四元数の出力 $y = iy_1 + jy_2 + ky_3$ を教師データとして行う。学習データは、振幅の1%のノイズを加えて作成したものを含めて78個用意した。学習の際の誤差の推移を Fig. 4 に示す。学習回数10000回で誤差は十分に減少している。

まず、通常の回転の場合の推定、すなわち x 軸、 y 軸、 z 軸周りの回転の場合の推定を行った。学習データと同様に作成したテストデータを用いた。Fig.5 に推定結果を示す。赤で示したのが y 軸周りの回転、緑が x 軸周り、青が z 軸周りの回転を表す。各軸周り $\pi/2$ 回転なので、円ではなく四角形となっているが、各軸周りの回転が学習・推定されていることが分かる。

次に、連続する回転の場合の推定、すなわち y 軸周りの回転に続いて z 軸周りの回転を行った場合と、 y 軸周りの回転に続いて x 軸周りの回転を行った場合の推定を行った。結果を Fig.6 に示す。青で示したのが y 軸 $\rightarrow z$ 軸の回転、赤が y 軸 $\rightarrow x$ 軸の回転である。これらより、2つの回転の合成回転が実現されている。また、ジンバルロックが生じていないこともわかる。

さらに、入力の筋電位に相当する四元数 $e = ie_1 + je_2 + ke_3$ の各部の値を $1/2$ にした場合の結果を Fig.7 に示す。Fig.5 に比べて移動のベクトル長が約半分になるとともに、ベクトルの方向が球面の接線方向となっていることから、球面補間が実現されている様子がわかる。

以上の結果より、筋電入力から前腕の3次元の回転が学習・推定できること、回転の合成によりジンバルロックが回避できること、球面補間が実現できることを示した。

5 おわりに

本研究では、筋電位からの前腕姿勢推定の問題を対象に、四元数ニューラルネットの導入を検討した。ニューラルネットの構成は、状態方程式に基づく動的モデルをもとにリカレント型とした。上腕二頭筋、円回内筋、棘下筋を想定した筋肉による前腕の回転をそれぞれ x, y, z 軸周りの回転としてとらえ、筋電位と回転の関係を四元数ニューラルネットで学習・推定することを試みた。本研究では単純化した筋電疑似波形および各軸周りの回転として計算機シミュレーションを行い、筋電位から前腕の3次元の回転が学習・推定できること、回転の合成によりジンバルロックが回避できること、球面補間が実現できることを示した。今後は実波形による学習・推定の検討や、姿勢や回転の量的関係についての検討、運動動作の解析やロボット制御への応用のための検討などを考えている。

Table 1 Network parameters

Parameters	Values
Number of input neurons	2
Number of hidden neurons	10
Number of output neurons	1
Learning rate ϵ	0.001
Momentum rate α	0.9
Number of training epoch	10000

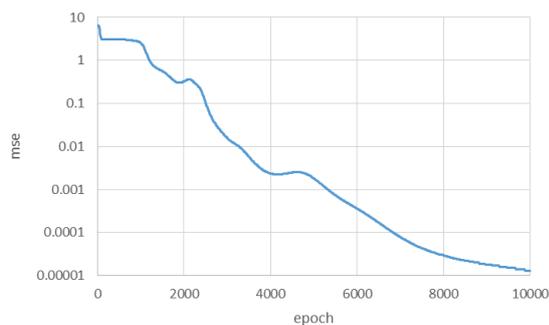


Fig.4 Convergence of learning by decreasing training error.

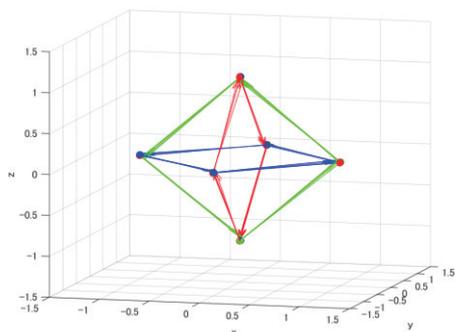


Fig.5 Result of estimation of rotation around each axis; green, red and blue arrows represent the rotations about the x axis, the y axis and the z axis, respectively.

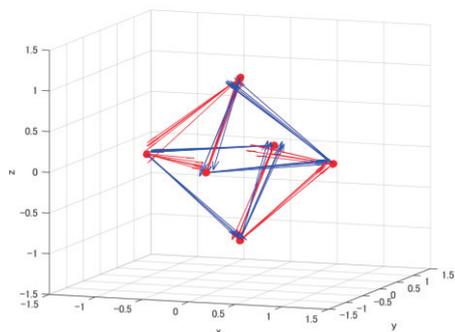


Fig.6 Result of estimation of two rotations; blue and red arrows indicate the rotations about the y to z axis and the y to x axis, respectively.

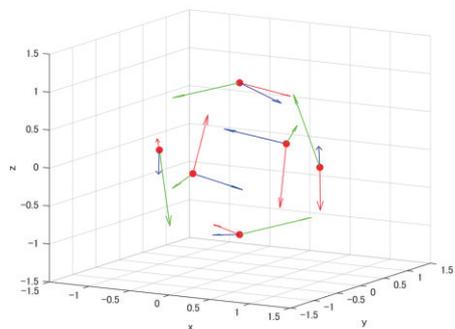


Fig. 7 Result of estimation of 1/2 rotation around each axis; green, red and blue arrows represent the rotations about the x axis, the y axis and the z axis, respectively.

謝辞

本研究の一部は科学研究費補助金（基盤 C，課題番号 26330284）の援助を受けて行われた。

参考文献

- 1) 金谷健一: 幾何学と代数系 Geometric Algebra -ハミルトン, グラスマン, クリフォード, 森北出版 (2014)
- 2) 金谷一郎: 3D・CG プログラマーのための実践クォータニオン, 工学社 (2004)
- 3) F. Dunn, I. Parberry, 松田晃一訳: 実例で学ぶゲーム 3D 数学, オライリージャパン (2008)
- 4) T. Nitta: An extension of the back-propagation algorithm to quaternions, Proc. of Int'l Conf. on Neural Information Processing, 1, 247/250 (1996)
- 5) N. Matsui, T. Isokawa, H. Kusamichi, F. Peper and H. Nishimura: Quaternion neural network with geometrical operators, Journal of Intelligent and Fuzzy Systems, 15, 3-4, 149/164 (2004)
- 6) Y. Kuroe: Models of Recurrent Clifford Neural Networks and Their Dynamics, in Complex-Valued Neural Networks: Advances and Applications (ed A. Hirose), John Wiley & Sons, 133/151 (2013)
- 7) 小川毅彦: 四元数ニューラルネットのロボットアーム逆運動学問題への応用, 第 8 回コンピューターショナル・インテリジェンス研究会講演論文集, 7/10 (2015)
- 8) 木塚, 木竜, 増田, 佐渡山: 表面筋電図, 東京電機大学出版局 (2006).
- 9) 赤沢堅造: 生体情報工学, 東京電機大学出版局 (2001)
- 10) 牧川, 吉田: 運動のバイオメカニクス, コロナ社 (2008)
- 11) M. F. Kelly, P. A. Parker, and R. N. Scott: Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study, IEEE Trans. on Biomedical Engineering, 37, 3, 221/230 (1990)
- 12) T. Tsuji, O. Fukuda, M. Kaneko, and K. Ito, "Pattern classification of time-series EMG signals using neural networks", Int. J. Adapt. Control Signal Process. 14, 829/848 (2000)
- 13) M. Hassan, A. Rahaman, F. Shuvo, A. S. Ovi, and M. Rahman: Human Hand Gesture Detection Based on EMG Signal Using ANN, Proc of the 3rd Int'l Conf. on Informatics, Electronics & Vision, (2014)
- 14) R. L. Drake, A. W. Vogl, and A. W. M. Mitchell (著), 塩田, 秋田(訳): グレイ解剖学 原著第 3 版, エルゼビアジャパン (2016)
- 15) H. Gray: Anatomy of the Human Body. (1918)