

深層展開を用いた静的出力フィードバック安定化における ハイパーパラメータの考察

和田 弘匡*・小蔵 正輝*

岸田 昌子**・若宮 直紀*

Hyperparameter Selection in Deep Unfolding-based Static Output Feedback Stabilization

Hirokata WADA*, Masaki OGURA*,
Masako KISHIDA** and Naoki WAKAMIYA*

The static output feedback stabilization problem has a basic and simple structure. However, this problem is NP-hard and difficult to solve. In this study, we apply the method of deep unfolding to design a static output feedback control system. We then investigate how the performance of algorithms based on deep unfolding for static output feedback stabilization depends on the hyperparameters of the optimizer. In particular, we provide a policy for choosing appropriate hyperparameters by evaluating the algorithms' stabilization success rates and learning times for various optimizers, learning rates, loss functions, and discretization periods.

Key Words: Deep Unfolding, Static Output Feedback Stabilization

1. はじめに

制御工学において制御系の安定性を保証することは重要な基本的な課題である。したがって制御理論においては、制御対象の性質、制御器の構造、そして安定性の種類に応じて様々な安定化問題が存在し、かつ盛んに研究されている^{1)~3)}。それら安定化問題の中で簡単な構造をもつものの一つに静的出力フィードバック安定化問題⁴⁾がある。この静的出力フィードバック安定化問題は、線形時不変システムをその出力を用いて静的なフィードバックゲインにより安定化する問題であり、極配置⁵⁾、遅れ系⁶⁾、量子化制御⁷⁾、確率制御⁸⁾、ファジィ制御⁹⁾など様々な問題設定で研究が行われている。一方で、静的出力フィードバック安定化問題はその単純な構造にもかかわらずNP困難であることが知られている^{10),11)}。この安定化問題に対して、線形行列不等式の反復的な求解に基づく手法¹²⁾や極配置法を用いた手法¹³⁾など様々なアルゴリズムが提案されている。

この状況に対して近年、深層展開^{14)~20)}と呼ばれる深層学習技術に由来する手法の静的出力フィードバック安定化問題

への適用が文献^{21),22)}で検討されており、一定の有効性が数値的に示唆されている。一方でこれらの文献では、深層展開にて重要な役割を果たすオプティマイザ²³⁾やハイパーパラメータに関する検討が十分になされていない。特に、与えられたシステムに対してハイパーパラメータをどのように選定すべきかが明らかではなく、そのために手法の使用性が必ずしも高くないという課題がある。

そこで本研究では、静的出力フィードバック安定化問題に対する深層展開に基づいたアルゴリズムの性能が、オプティマイザや学習率などのハイパーパラメータに対してどのように依存するかを調査し考察する。特に本稿では1) オプティマイザ、2) 学習率、3) 損失関数、および4) 離散化周期を様々な変化させたときの、安定化成功率や学習回数を数値実験により調べることを通じて、適切なハイパーパラメータの決定方針を与える。

本稿の構成は以下の通りである。第二章では本稿で扱う制御問題である静的出力フィードバック安定化問題や、深層展開を用いた制御器設計アルゴリズム、およびそのアルゴリズムで使用されるオプティマイザについて述べる。第三章ではアルゴリズム性能のハイパーパラメータに対する依存性を考察する。第四章では考察を踏まえた結論を述べる。

2. 問題設定

2.1 静的出力フィードバック安定化問題 連続時間線形時不変システム

* 大阪大学大学院情報科学研究科 大阪府吹田市山田丘 1-5

** 国立情報学研究所 情報学プリンシプル研究系 東京都千代田区一ツ橋 2-1-2

* Graduate School of Information Science and Technology, Osaka University, Suita-shi, Osaka

** National Institute of Informatics, Tokyo

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t)\end{aligned}\quad (1)$$

を考える．ここで $x(t) \in \mathbb{R}^n$ は状態， $y(t) \in \mathbb{R}^p$ は出力， $u(t) \in \mathbb{R}^m$ は入力である．係数行列 A, B, C は定数であり，かつ制御系の設計者にとって既知であるとする．設計する制御器は静的な出力フィードバック

$$u(t) = Ky(t) \quad (2)$$

を考える．結果として得られる閉ループ系は

$$\Sigma_K: \dot{x}(t) = (A + BKC)x(t) \quad (3)$$

である．このシステムの安定性は以下のように定義される．

[定義] システム(3)が安定であるとは，任意の初期状態 $x(0) \in \mathbb{R}^n$ に対して $\lim_{t \rightarrow \infty} x(t) = 0$ が成り立つことを言う．

静的出力フィードバック安定化問題は以下のように定式化される．

[問題 1] 閉ループ系(3)が安定であるようなフィードバックゲイン K を求めよ．

2.2 制御器設計アルゴリズム

本項では静的出力フィードバック安定化のための深層展開に基づいたアルゴリズムを述べる．

深層展開は，誤差逆伝播法や確率的勾配法などの深層学習技術を利用した反復アルゴリズムの改善手法である^{14)~20)}．時間方向に展開された信号流グラフを深層ニューラルネットワークとみなすことで，グラフに埋め込まれた学習可能パラメータの調整が可能となる．また，深層展開は離散時間のアルゴリズムを対象とした手法であり，本稿で考える連続時間における静的出力フィードバック安定化問題に対してそのまま適用することができない．そこで本稿では閉ループ系(3)を周期 $h > 0$ で以下のように離散化する：

$$x((k+1)h) = \exp((A + BKC)h)x(kh). \quad (4)$$

また，深層展開を適用するために損失関数と呼ばれる指標を準備する必要がある．本稿では文献²²⁾と同様に静的出力フィードバック安定化問題を有限時間における状態のノルムの最小化問題とみなすことで，次の2つの損失関数

$$L_{\log}(t; x_0) = \log \|x(t)\| \quad (5)$$

$$L_{\text{MSE}}(t; x_0) = \|x(t)\|^2 \quad (6)$$

を用いる．式(5)では文献²²⁾とは異なりノルム $\|x(t)\|$ の対数をとっていることに注意する．これは後述のフィードバックゲインの学習中における数値的な不安定性を避けるためである．

差分方程式(4)を反復アルゴリズムとみなせば，深層展開の考え方を適用することにより静的出力フィードバック安定化のための以下のようなアルゴリズム²²⁾が得られる．

(1) 離散化の周期 $h > 0$ ，フィードバックゲイン K の初

期値，自然数 M ，正の数 T を定める．空間 \mathbb{R}^n における確率分布を定める．

(2) ステップ1で定めた確率分布から M 個の初期状態 x_0 をランダムに生成する．そして $L(h; x_0)$ を損失関数として確率的勾配降下法によりフィードバックゲイン K の学習を行う．ここでのバッチサイズは1である．学習終了時のフィードバックゲインを $K^{(1)}$ とする．このゲインを用いた閉ループ系 $\Sigma_{K^{(1)}}$ が漸近安定であればアルゴリズムを終了する．

(3) 任意の $k \geq 2$ に対して，損失関数を $L(kh; x_0)$ ，初期値を $K^{(k-1)}$ として確率的勾配降下法により K の学習を行う．初期状態 x_0 はステップ2と同様の方法で生成する．学習終了時のフィードバックゲイン K を $K^{(k)}$ とする．閉ループ系 $\Sigma_{K^{(k)}}$ が漸近安定であればアルゴリズムを終了する．

(4) ステップ3は $k = T$ まで繰り返される．

以下に，深層展開に基づく他のアルゴリズム^{21), 22)}と，本稿で提案するアルゴリズムの違いを述べる．一つ目が損失関数(5)における対数の使用である．対数の導入によりアルゴリズム実行時の数値的なオーバーフローの発生頻度を下げることができるが期待される．二つ目は離散化(4)の方法である．文献²¹⁾で用いられている双一次変換と比べて，本稿で用いている状態のサンプリングはより自然な離散化だと考えられる．

2.3 オプティマイザ

本項では，深層展開で使用する最適化アルゴリズムであるオプティマイザ²³⁾のうち，本研究で用いたSGD, Adagrad, Adadelata, RMSprop, Adamの特徴を述べる．

2.3.1 SGD

SGDは最急降下法に改良を加えたアルゴリズムであり，最急降下法において極小値に陥ってしまうことを解決する．しかし，学習率の選択が困難である．小さすぎる学習率は収束を極端に遅くし，大きすぎる学習率は収束を妨げてしまうからである．また，全てのパラメータ更新に同じ学習率が適用されてしまうため，データが疎で特徴の出現頻度が非常に異なる場合，全ての特徴を同じように更新してしまうため，出現頻度が少ない特徴を見逃してしまう可能性がある．

2.3.2 Adagrad

AdagradはSGDにおける出現頻度が少ない特徴を見逃してしまうという欠点を改善するアルゴリズムである．しかし，学習が進むにつれて学習率を小さくするため，最終的には学習率が微小になってしまい，その時点で学習が終わってしまうという欠点がある．学習率の推奨値は 10^{-2} である．

2.3.3 Adadelata

AdadelataはAdagradの最終的に学習が終わってしまうという欠点を改善したアルゴリズムである．また，更新における勾配の単位を揃える工夫をしたアルゴリズムである．Adadelataには学習率がないため，設定する必要がない．

2.3.4 RMSprop

RMSpropもAdagradの最終的に学習が終わってしまうと

いう欠点を改善したアルゴリズムである。このオプティマイザでは、最近の学習ほど強く影響を与えるように工夫されている。学習率の推奨値は 10^{-3} である。

2.3.5 Adam

Adam は SGD に勾配の加速度を加えたアルゴリズムである momentumSGD と RMSprop を組み合わせたアルゴリズムである。勾配と勾配の 2 乗の情報をため込むことで、2 つのアルゴリズムの組み合わせを実現している。学習率の推奨値は 10^{-3} である。

3. アルゴリズム性能のハイパーパラメータに対する依存性

前節で述べた制御器設計アルゴリズムの性能を評価する。特に本稿では、性能のオプティマイザや学習率などのハイパーパラメータに対する依存性を調べる。本節ではオプティマイザ、学習率、 h を変化させ、アルゴリズム性能の依存性を、成功率と学習にかかった回数を用いて評価する。

性能評価を行う際に必要なシステム(1)は以下の手順¹²⁾でランダムに 100 個生成する。まず、固有値の実部の最大値が 0 となる行列 \tilde{A} をランダムに生成する。この時の \tilde{A} は、 -1 から 1 の範囲で一様乱数を用いて生成された行列を P 、その行列の固有値の実数部の最大値を α とし、 $\tilde{A} = P - \alpha I$ により定める。また行列 K, B, C を各要素について、 -1 から 1 の範囲で一様乱数を用いて生成する。そして行列 A を $A = \tilde{A} - BKC$ により定める。この方法で生成された行列 A, B, C を係数行列として持つシステム(1)に対しては、閉ループ系のシステム行列 $A + BKC = \tilde{A}$ の固有値の実部が 0 以下になるようなフィードバックゲイン K の存在が保証されている。

アルゴリズムの性能評価で共通して用いるパラメータを述べる。学習アルゴリズムにおけるパラメータ M, T はそれぞれ $M = 400, T = 600$ で固定する。離散化幅 h は 10^{-3} から 10^1 まで対数スケールで変化させる。状態空間 \mathbb{R}^n における確率分布は積集合 $[-1, 1]^n$ 上の一様分布とする。学習開始時におけるフィードバックゲイン K の初期値は零行列とする。確率的勾配降下法におけるステップサイズの調整には 2.3 で述べた SGD, Adagrad, Adadelta, RMSprop, Adam を用いる。状態 x の次元 n , 入力 u の次元 m , 出力 y の次元 p の組み合わせは、小規模, 中規模, 大規模を想定して $(n, m, p) = (4, 1, 1), (8, 2, 2), (12, 3, 3)$ とする。各オプティマイザにおける学習率 η は 10^{-5} から 10^5 まで対数スケールで変化させる。Adadelta について、本来学習率は存在しないが、今回使用する pytorch において探索幅の初期値のパラメータとして学習率が用意されているため、Adadelta においても学習率を変化させる。本来の Adadelta となるための学習率は 1 である。

まず $(n, m, p) = (4, 1, 1)$ における各ハイパーパラメータに対する依存性を評価する。Fig.1 に、各オプティマイザにおける学習率と h を変化させたときの成功率が 95%以上となるハイパーパラメータの組み合わせの範囲を示す。本研究で

は、成功率が 95%以上となるハイパーパラメータを適切なハイパーパラメータと定義する。Fig.1 から、以下の 3 つの特徴がわかる。

- h が 1 や 10 の時、同じオプティマイザにおいて MSE と log で比較すると log のほうが適切な学習率の範囲が広い
- h が 10^{-3} のとき、全てのオプティマイザにおいて適切な学習率の範囲が最も狭い
- SGD では log と MSE 両方で、 h と適切な学習率に反比例関係がある

以下では上記特徴の理由を考察する。 h が 1 や 10 の時、同じオプティマイザにおける MSE と log で比較すると log のほうが適切な学習率の範囲が広いのは、MSE では学習中に損失の値が nan となりプログラムがエラー終了してしまうからである。これは、 h が大きいと十分に K が学習されていない状態で t が大きい時の x を計算してしまい、その結果 nan が出てしまうと考えられる。また、 h が 10^{-3} のとき全てのオプティマイザにおいて適切な学習率の範囲がせまいのは、 h の小ささに対して T が小さいために十分な時間幅での学習が行えないため、安定化できる K を見つけることができていないと考えられる。最後に、SGD において log と MSE 両方で h と適切な学習率が反比例関係にあるのは、SGD は学習率の大小がそのまま学習する際の探索幅を表しており、その探索幅が学習の進み具合に応じて変化しないからであると考えられる。

また、Fig.2 に各オプティマイザにおける学習にかかる回数の平均値を表すグラフを示す。ただし、学習率が 95%以上とまらない成功率については省いている。全てのオプティマイザの場合で、 h が大きいほど同じ学習率において学習回数が少なくなる傾向がある。また SGD においては、log, MSE ともに学習回数と学習率の間に負の相関がある。SGD 以外の 4 つのオプティマイザについて、 h が大きい時、同じ学習率において log は MSE よりも学習回数が少ないことがわかる。

以上のことから、 $(n, m, p) = (4, 1, 1)$ においては、損失関数 log かつ、 h は 10^1 を使うべきであることがわかる。また、最も学習回数が少なくなるような学習率がわからない学習前の状況において、使用するオプティマイザは損失関数 log, $h = 10^1$ で成功率が 95%以上となる学習率の範囲が広く学習回数も少ない Adadelta や RMSprop を選ぶべきであることがわかる。 $(n, m, p) = (4, 1, 1)$ においては、損失関数 log, $h = 10^1$, Adadelta の学習率 10^2 や RMSprop の学習率 10^{-1} が最も優れているが、システムの規模が大きくなった場合や他の制御問題においても同様の結果が得られるかを確認する必要がある。

次に $(n, m, p) = (8, 2, 2)$ における各ハイパーパラメータへの成功率の依存性を評価する。Fig.3 に、各オプティマイザにおける学習率と h を変化させたときの成功率が 95%以上となるハイパーパラメータの組み合わせの範囲を表すグラフを示す。 $(n, m, p) = (4, 1, 1)$ と比較して、一部適切な学習率の範囲が狭くなっている部分はあるが、先ほど述べた 3 つの特

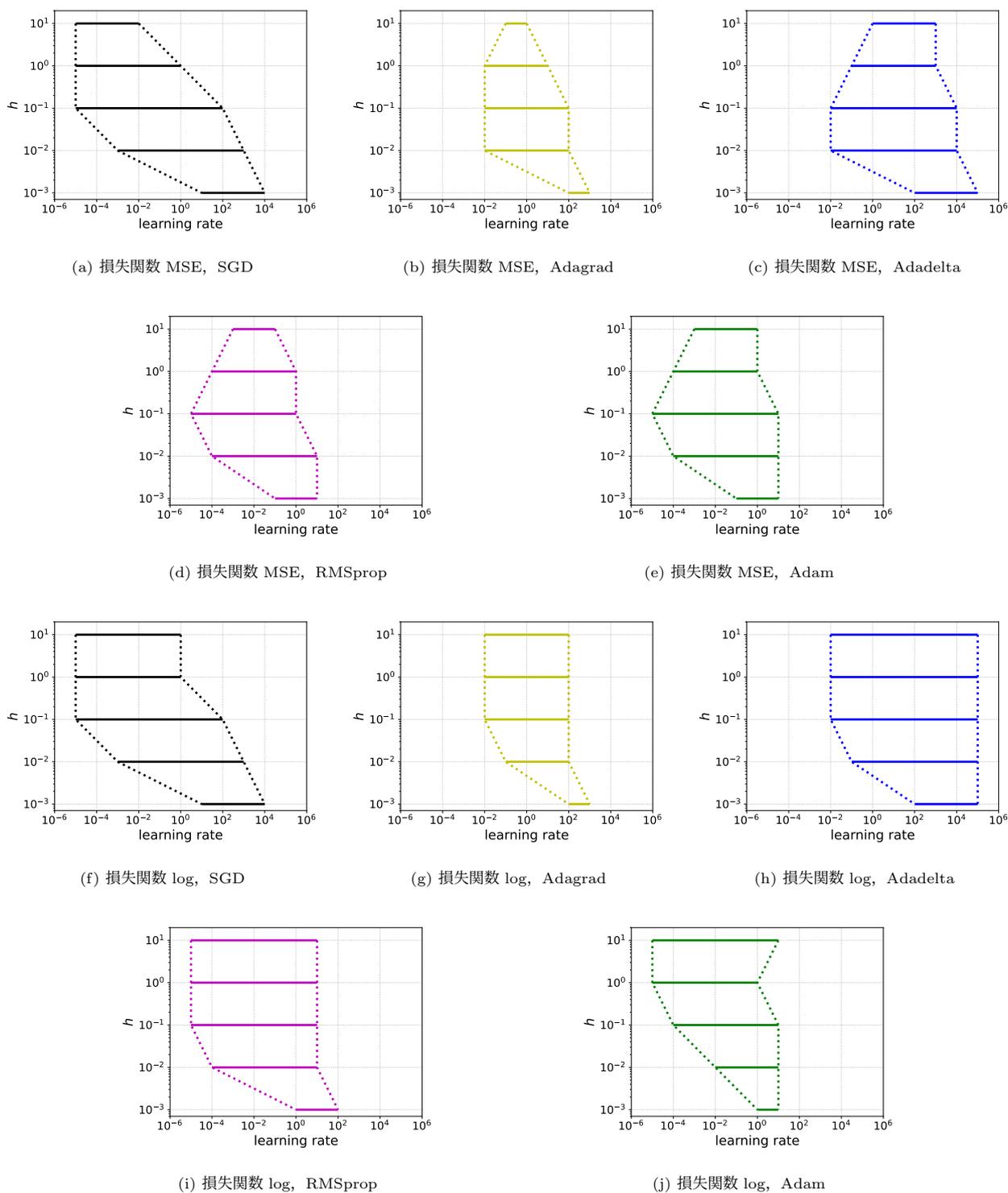


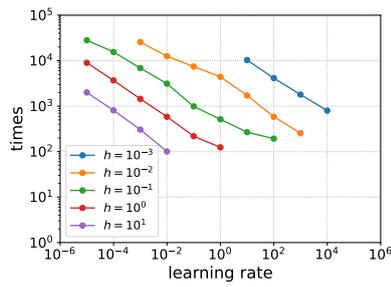
Fig. 1: $(n, m, p) = (4, 1, 1)$ における成功率が 95%以上となる範囲

徴が $(n, m, p) = (8, 2, 2)$ においても見られた。

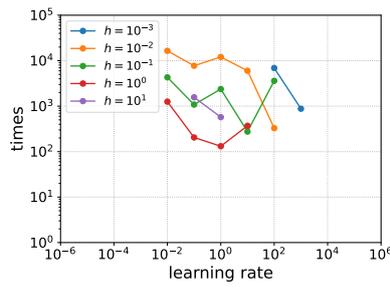
また, Fig.4 に $(n, m, p) = (8, 2, 2)$ における各オプティマイザにおける学習にかかる回数の平均値を表すグラフを示す。 $(n, m, p) = (4, 1, 1)$ と比較して, $(n, m, p) = (8, 2, 2)$ においても同様の特徴が見られた。しかし, 損失関数 log, $h = 10^1$, Adadelta や RMSprop で成功率が 95%以上とな

る範囲が少し狭くなった。 $(n, m, p) = (4, 1, 1)$ と同様に, $(n, m, p) = (8, 2, 2)$ においても損失関数 log, $h = 10^1$, Adadelta の学習率 10^2 や RMSprop の学習率 10^{-1} が最も優れていた。

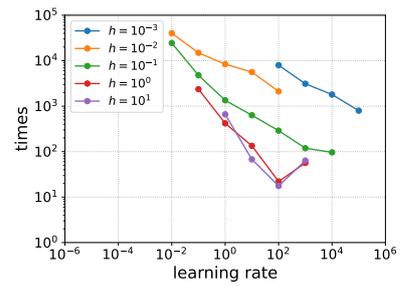
最後に $(n, m, p) = (12, 3, 3)$ における各ハイパーパラメータの依存性を評価する。 Fig.5 に, 各オプティマイザにおけ



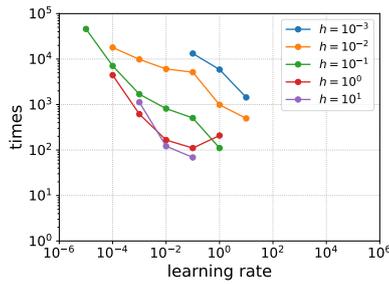
(a) 損失関数 MSE, SGD



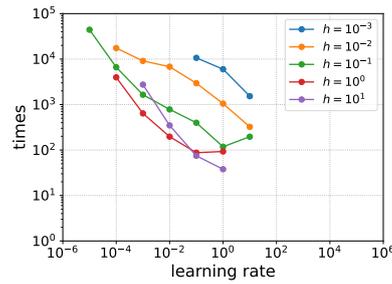
(b) 損失関数 MSE, Adagrad



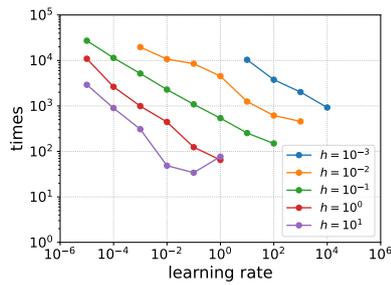
(c) 損失関数 MSE, Adadelta



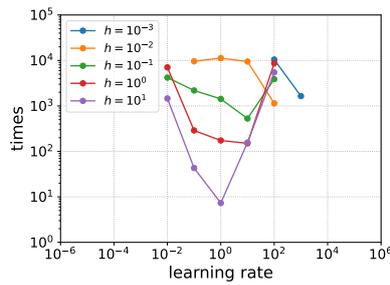
(d) 損失関数 MSE, RMSprop



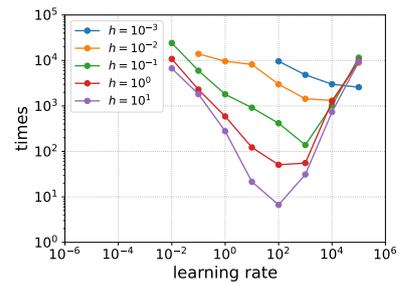
(e) 損失関数 MSE, Adam



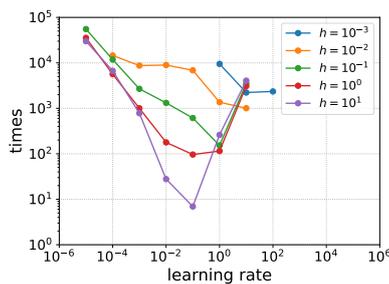
(f) 損失関数 log, SGD



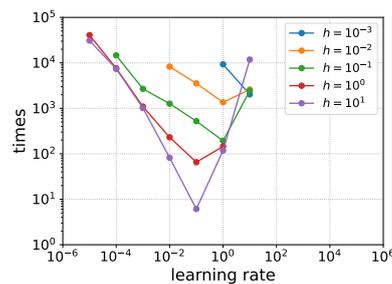
(g) 損失関数 log, Adagrad



(h) 損失関数 log, Adadelta



(i) 損失関数 log, RMSprop



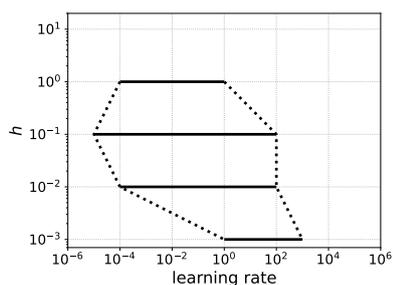
(j) 損失関数 log, Adam

Fig. 2: $(n, m, p) = (4, 1, 1)$ における学習にかかる回数

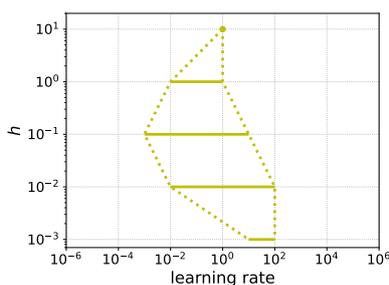
る学習率と h を変化させたときの成功率が 95%以上となるハイパーパラメータの組み合わせの範囲を表すグラフを示す。 $(n, m, p) = (4, 1, 1), (8, 2, 2)$ と比較して、一部適切な学習率の範囲が変化している部分はあるが、同じ特徴が $(n, m, p) = (12, 3, 3)$ においても見られた。

また、Fig.6 に $(n, m, p) = (12, 3, 3)$ における各オプティマ

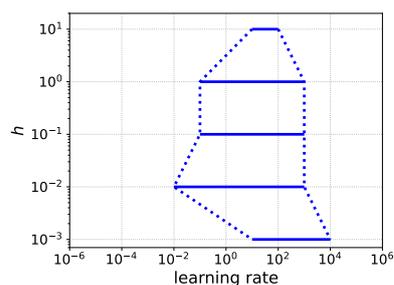
イザにおける学習にかかる回数の平均値を表すグラフを示す。 $(n, m, p) = (8, 2, 2)$ と比較して、 $(n, m, p) = (12, 3, 3)$ においても同様の特徴が見られた。 $(n, m, p) = (4, 1, 1), (8, 2, 2)$ と同様に、 $(n, m, p) = (12, 3, 3)$ においても損失関数 log, $h = 10^1$, Adadelta の学習率 10^2 や RMSprop の学習率 10^{-1} が最も優れていた。



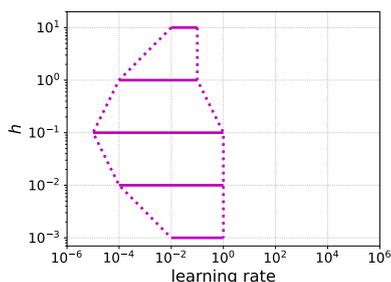
(a) 損失関数 MSE, SGD



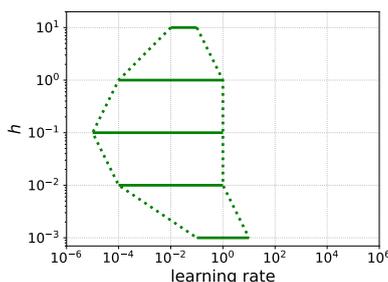
(b) 損失関数 MSE, Adagrad



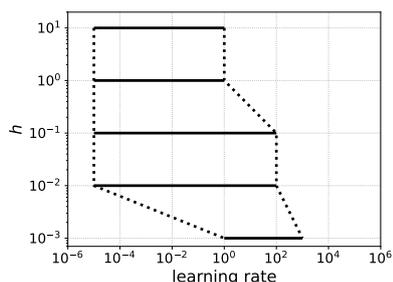
(c) 損失関数 MSE, Adadelta



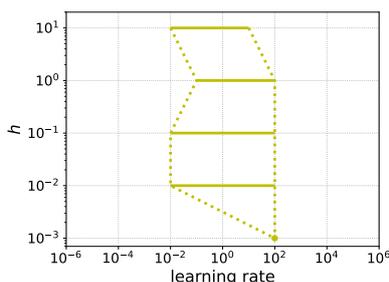
(d) 損失関数 MSE, RMSprop



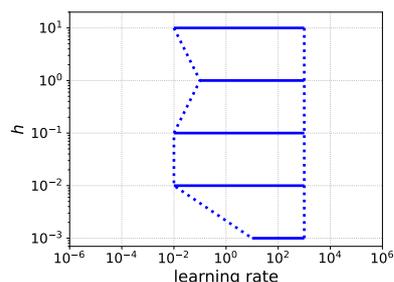
(e) 損失関数 MSE, Adam



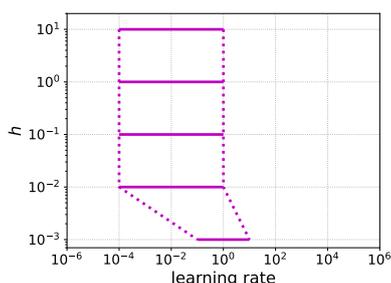
(f) 損失関数 log, SGD



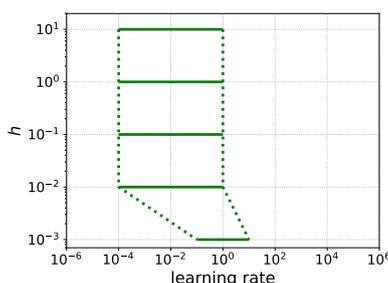
(g) 損失関数 log, Adagrad



(h) 損失関数 log, Adadelta



(i) 損失関数 log, RMSprop

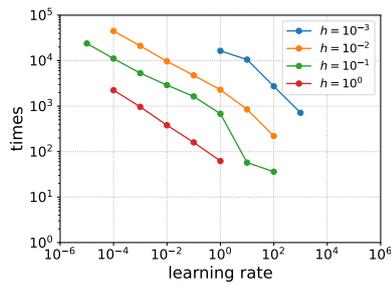


(j) 損失関数 log, Adam

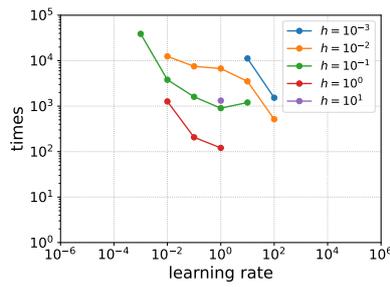
Fig. 3: $(n, m, p) = (8, 2, 2)$ における成功率が 95%以上となる範囲

全ての規模において、各オプティマイザについて1つずつ最適な学習率を確認していくと、SGDでは先ほど述べたように h と学習率の間に相関があり、全ての h に対して学習回数が少なくなるような学習率は存在しない。Adagradでは、学習率の初期パラメータとして 10^{-2} が与えられているが、今回の条件下ではより大きい 10^0 や 10^1 のほうが学習回数が少

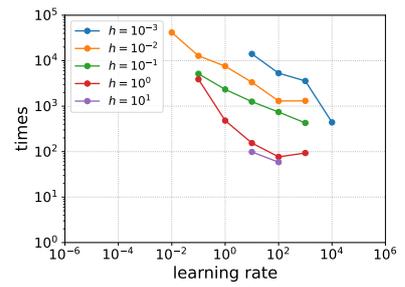
なくなる傾向が見られた。Adadeltaは学習率 10^0 が本来の最適化アルゴリズムであるが、今回の条件下では 10^3 のときに最も学習回数が少なくなっている。RMSpropは、学習率の初期パラメータとして 10^{-3} が与えられているが、本研究では 10^{-1} で最も学習回数が少なくなっている。Adamでは学



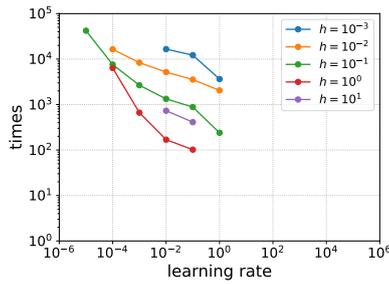
(a) 損失関数 MSE, SGD



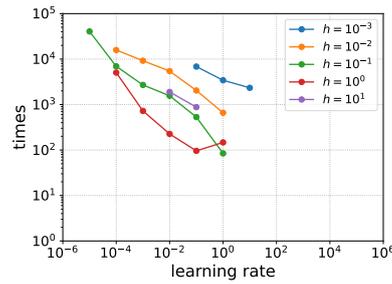
(b) 損失関数 MSE, Adagrad



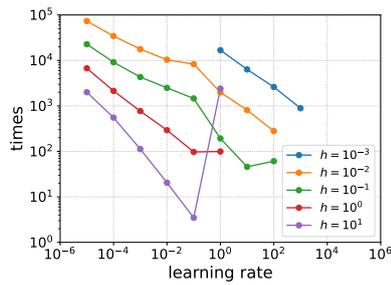
(c) 損失関数 MSE, Adadelta



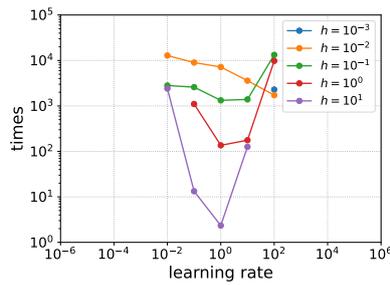
(d) 損失関数 MSE, RMSprop



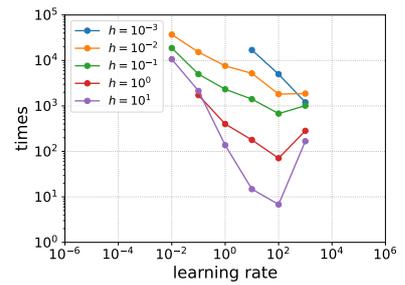
(e) 損失関数 MSE, Adam



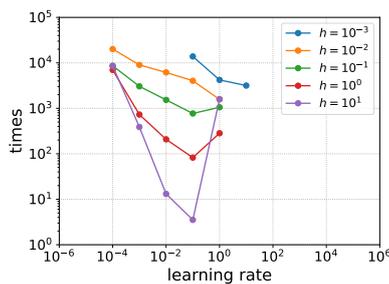
(f) 損失関数 log, SGD



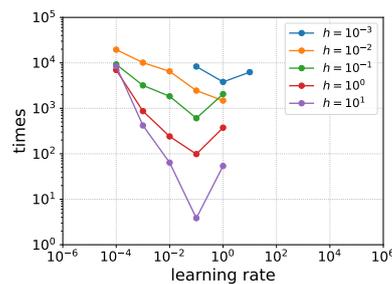
(g) 損失関数 log, Adagrad



(h) 損失関数 log, Adadelta



(i) 損失関数 log, RMSprop

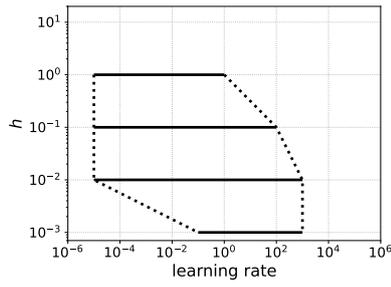


(j) 損失関数 log, Adam

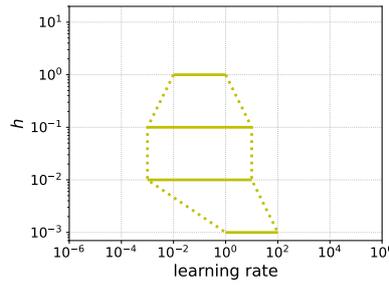
Fig. 4: $(n, m, p) = (8, 2, 2)$ における学習にかかる回数

習率の初期パラメータとして 10^{-3} が与えられているが、本研究では 10^{-1} の時に最も学習回数が少なくなっている。つまり本研究の条件下では、SGD 以外の 4 つのオプティマイザについて、初期パラメータよりも大きい学習率において最も学習回数が少なくなっていた。学習率に関するこの結果が、

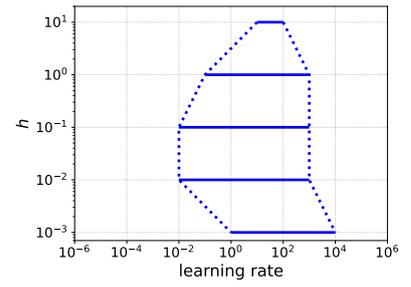
静的出力フィードバック問題以外の制御問題に対して深層展開を適用した場合においても成り立つかどうかの検討は本稿では行わない。しかしながら、静的出力フィードバック問題に帰着可能であることが知られている様々な最適制御問題²⁴⁾ やあるクラスの動的出力フィードバック安定化問題⁴⁾ においては、上述のように学習率を初期パラメータよりも大きくす



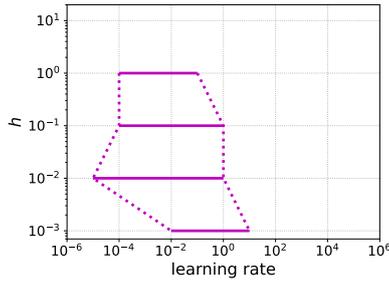
(a) 損失関数 MSE, SGD



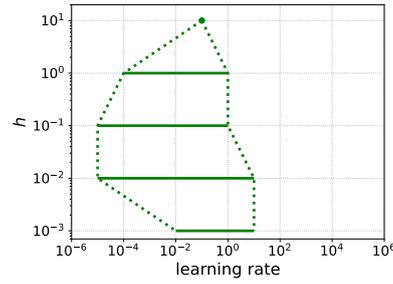
(b) 損失関数 MSE, Adagrad



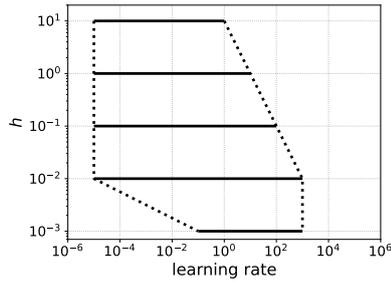
(c) 損失関数 MSE, Adadelta



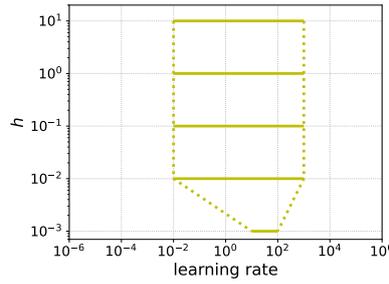
(d) 損失関数 MSE, RMSprop



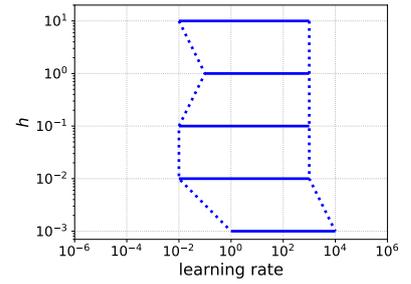
(e) 損失関数 MSE, Adam



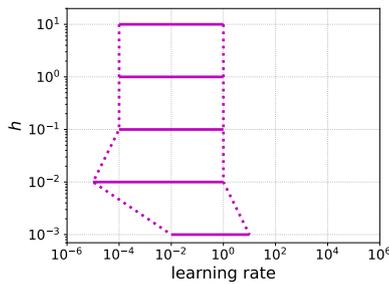
(f) 損失関数 log, SGD



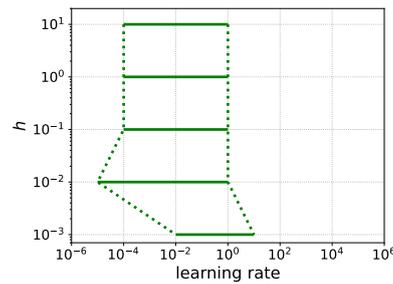
(g) 損失関数 log, Adagrad



(h) 損失関数 log, Adadelta



(i) 損失関数 log, RMSprop



(j) 損失関数 log, Adam

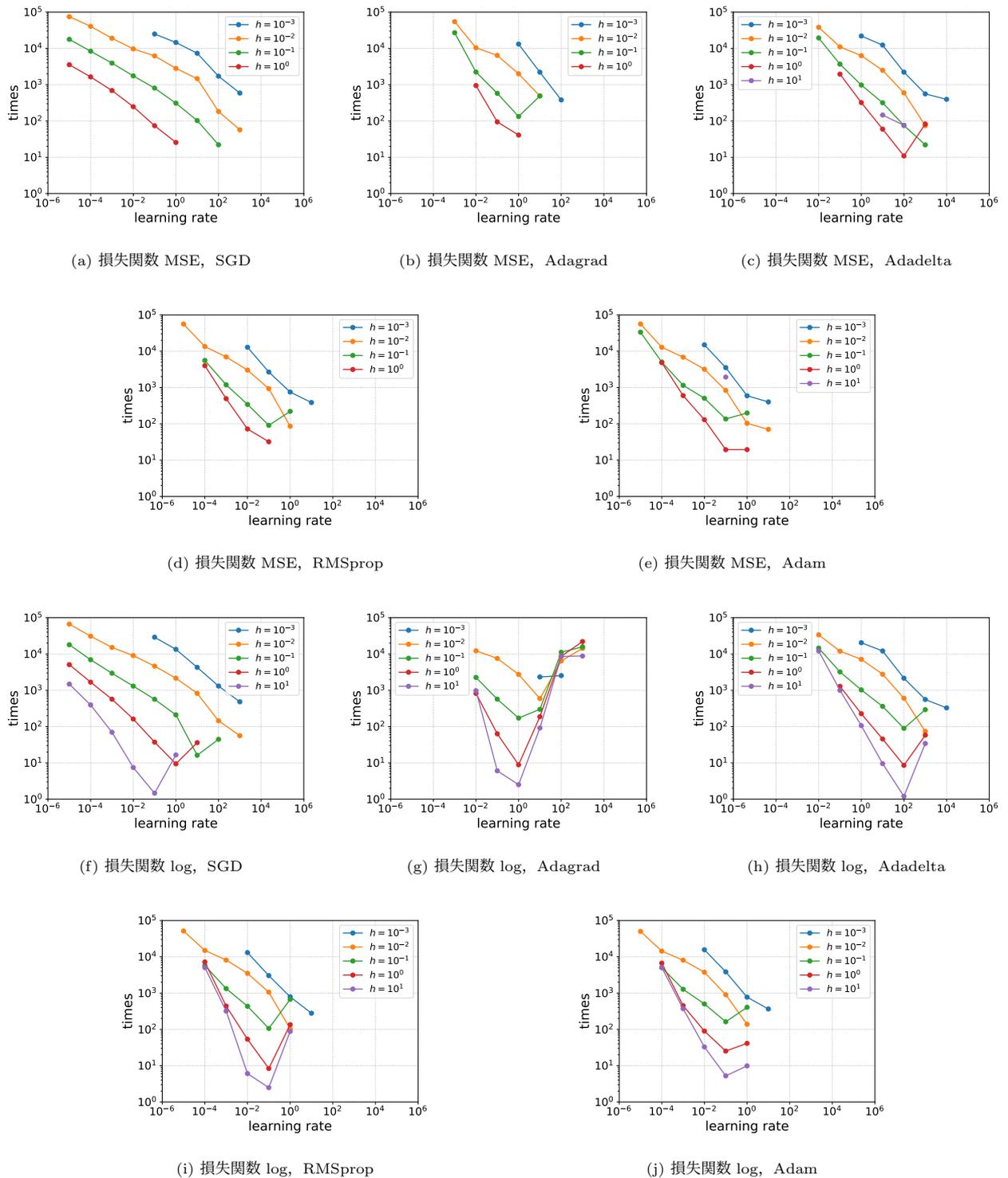
Fig. 5: $(n, m, p) = (12, 3, 3)$ における成功率が 95%以上となる範囲

の方が良いということが期待される。

以上のことから、本研究における条件下ではシステムの規模に関わらず、損失関数 log, $h = 10^1$, Adadelta の学習率 10^2 や RMSprop の学習率 10^{-1} が最も優れていることがわかった。今後は、他の制御問題においても同様の特徴が得られるかを検証する必要がある。

4. 結 論

本研究では静的出力フィードバック安定化問題に対して深層展開を適用した際の、アルゴリズム性能の最適化や学習率などのハイパーパラメータに対する依存性について考察を行った。システムの規模を3段階用意し、損失関数、オブ

Fig. 6: $(n, m, p) = (12, 3, 3)$ における学習にかかる回数

ティマイザ, 学習率, 時間幅 h をさまざまに変化させてシミュレーションを行った結果, 損失関数 log, $h = 10$, RMSprop における学習率 10^{-1} , または Adadelta における学習率 10^2 が最も優れていることがわかった.

しかし, 本研究の結果は特定の条件を満たすシステム (3) の静的出力フィードバック安定化問題において得られた結果

であり, 他の条件や制御問題においても同様のハイパーパラメータが優れているかは不明である. そこで今後の課題として, 他の条件や制御問題においてもハイパーパラメータの調査が必要であると考えられる.

謝辞 本研究は JST CREST (課題番号: JPMJCR2012), 大阪大学大学院情報科学研究科スタートアッププログラム, お

および JSPS 科研費 JP21H01352, JP21H01353, JP22H00514 の助成を受けた。ここに謝意を表す。

利益相反 利益相反に該当する事項はありません。

責任著者 小蔵正輝, m-ogura@ist.osaka-u.ac.jp, 大阪大学大学院情報科学研究科

著者の貢献 和田弘匡：概念化・調査・方法論・ソフトウェア・可視化。執筆-原稿作成。小蔵正輝：概念化・資金獲得・方法論・プロジェクト管理・指導・執筆-レビューおよび編集校正。岸田昌子：資金獲得・考察・執筆-レビューおよび編集校正。若宮直紀：指導・執筆-レビューおよび編集校正。

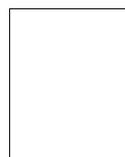
参考文献

- 1) W. P. M. H. Heemels, and N. van de Wouw, “Stability and stabilization of networked control systems,” *Networked Control Systems*, pp. 203–253, 2010.
- 2) E. D. Sontag, “Feedback stabilization of nonlinear systems,” *Robust Control of Linear Systems and Nonlinear Control*, vol. 4, pp. 61–81, 1990.
- 3) H. Lin, and P. J. Antsaklis, “Stability and stabilizability of switched linear systems: a survey of recent results,” *IEEE Transactions on Automatic Control*, vol. 54, pp. 308–322, 2009.
- 4) V. Szymos, C. Abdallah, P. Dorato, and K. Grigoriadis, “Static output feedback—A survey,” *Automatica*, vol. 33, pp. 125–137, 1997.
- 5) A. Eremenko and A. Gabrielov, “Pole placement by static output feedback for generic linear systems,” *SIAM Journal on Control and Optimization*, vol. 41, pp. 303–312, 2002.
- 6) V.L. Kharitonov, S.-I. Niculescu, J. Moreno and W. Michiels, “Static output feedback stabilization: necessary conditions for multiple delay controllers,” *IEEE Transactions on Automatic Control*, vol. 50, pp. 82–86, 2005.
- 7) X. H. Chang, J. Xiong, Z. M. Li and J. H. Park, “Quantized static output feedback control for discrete-time systems,” *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 3426–3435, 2018.
- 8) H. Mukaidani, H. Xu and V. Dragan, “Stochastic optimal control for weakly coupled large-scale systems via state and static output feedback,” *IET Control Theory & Applications*, vol. 4, pp. 1849–1858, 2010.
- 9) J. Yoneyama and K. Hoshino, “Static output feedback control design for Takagi-Sugeno descriptor fuzzy systems,” In *2015 International Conference on Informatics, Electronics & Vision*, 2015.
- 10) V. Blondel and J. N. Tsitsiklis, “NP-hardness of some linear control design problems,” *SIAM Journal on Control and Optimization*, vol. 35, pp. 2118–2127, 1997.
- 11) B. T. Polyak and P. S. Shcherbakov, “Hard problems in linear control theory: Possible approaches to solution,” *Automation and Remote Control*, vol. 66, pp. 681–718, 2005.
- 12) Y. Y. Cao, J. Lam, and Y. X. Sun, “Static output feedback stabilization: An ILMI approach,” *Automatica*, vol. 34, pp. 1641–1645, 1998.
- 13) 不破 勝彦, 成清 辰生, 大羽 達志, “静的出力フィードバック制御における一考察,” *電気学会論文誌 C (電子・情報・システム部門誌)*, vol. 141, pp. 496–497, 2021.
- 14) D. Ito, S. Takabe, and T. Wadayama, “Trainable ISTA for sparse signal recovery,” *IEEE Transactions on Signal Processing*, vol. 67, pp. 3113–3125, 2019.

- 15) 和田山正, 高邊賢史, “深層展開に基づく信号処理アルゴリズムの設計—収束加速とその理論的解釈—,” *電子情報通信学会基礎・境界ソサイエティ Fundamentals Review*, vol. 14, pp. 60–72, 2020.
- 16) V. Monga, Y. Li and Y. C. Eldar, “Algorithm unrolling: Interpretable efficient deep learning for signal and image processing,” *IEEE Signal Processing Magazine*, vol.38, pp. 18–44, 2021.
- 17) A. Jagannath, J. Jagannath and T. Melodia, “Redefining wireless communication for 6G: Signal processing meets deep learning with deep unfolding,” *IEEE Transactions on Artificial Intelligence*, vol.2, pp. 528–536, 2021.
- 18) 相澤純平, 小蔵正輝, 岸田昌子, 若宮直紀, “時相深層展開を用いたモデル予測制御の多重振り子系に対する有効性の検証,” *システム制御情報学会論文誌 (採録決定)*, 2022.
- 19) M. Kishida and M. Ogura, “Temporal deep unfolding for constrained nonlinear stochastic optimal controls,” *IET Control Theory & Applications*, vol. 16, no. 2, pp. 139–150, 2022.
- 20) M. Ogura, K. Kobayashi, and K. Sugimoto, “Static output feedback synthesis of time-delay linear systems via deep unfolding,” In *17th IFAC Workshop on Time Delay Systems*, pp. 214–215, 2022.
- 21) 小林恒輝, 小蔵正輝, 岸田昌子, 和田山正, 杉本謙二, “深層展開による出力フィードバック安定化の検討,” *信学技法*, vol. 119, pp. 59–62, 2019.
- 22) —, “Neural Ordinary Differential Equation を用いた静的出力フィードバック安定化の検討,” *信学技法*, vol. 119, pp. 19–22, 2020.
- 23) S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- 24) C. N. Nett, D. S. Bernstein and W. M. Haddad, “Minimal complexity control law synthesis, part I: Problem formulation and reduction to optimal static output feedback,” In *1989 American Control Conference*, 1989.

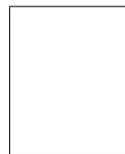
[著者紹介]

和田 弘匡



2020年大阪大学基礎工学部卒業。大阪大学大学院情報科学研究科在学中。深層展開を用いた制御系設計に関する研究に従事。

小蔵 正輝 (正会員)



2014年テキサス工科大学博士課程修了。Ph.D. (Mathematics)。ペンシルベニア大学博士研究員、奈良先端科学技術大学院大学助教を経て、2019年より大阪大学情報科学研究科准教授。現在に至る。専門分野はネットワーク科学、動的システム、およびそれらの最適化。2012年計測自動制御学会論文賞。2018年計測自動制御学会制御部門制御部門大会賞。2019年 IEEE Transactions on Network Science and Engineering Best Paper Award (Runners-up)。2021年計測自動制御学会著述賞。

岸田 昌子



2010 年イリノイ大学アーバナシャンペーン校博士課程修了, Ph.D. マサチューセッツ工科大学, 東京大学, カンタベリー大学, マグデブルグ大学を経て, 2016 年より国立情報学研究所および総合研究大学院大学准教授. 2021 年より大阪大学招へい准教授 (兼任). 2022 年より JST さきがけ研究員 (兼任). 専門分野は不確かさの解析, ネットワーク化制御理論および最適化. 2015 年フンボルトリサーチフェロー, 2018 年度電気通信普及財団テレコムシステム技術賞, 令和 2 年度科学技術分野の文部科学大臣表彰若手科学者賞.

若宮 直紀



1996 年大阪大学大学院基礎工学研究科博士後期課程修了 (物理系専攻). 博士 (工学). 大阪大学助手, 助教授, 准教授 (職名変更) を経て, 2011 年より大阪大学大学院情報科学研究科教授. 現在に至る. 専門分野は脳や生体に学ぶ情報通信技術, 自己組織化ネットワーク制御.

.....