

リザーバーコンピューティングを用いたホールスラストの推進性能予測モデルに関する研究

伊藤, 寛晟
九州大学大学院総合理工学府総合理工学専攻プラズマ・量子理工学メジャー

<https://hdl.handle.net/2324/6792859>

出版情報 : Kyushu University, 2022, 修士, 修士
バージョン :
権利関係 :



令和 4 年度
九州大学大学院 総合理工学府
総合理工学専攻
プラズマ・量子理工学メジャー

修 士 論 文

論文名

リザーバーコンピューティングを用いた
ホールスラスタの推進性能予測モデルに関する
研究

氏 名

伊藤 寛晟

指導教員名

山本 直嗣 教授

目次

第1章	序論.....	3
1-1.	研究背景.....	3
1-2.	ホールスラスト.....	3
1-2-1	作動原理および基本設計.....	3
1-2-2	放電電流の非線形性.....	5
1-2-3	ホールスラストの寿命.....	6
1-3.	研究目的.....	7
第2章	リザーバーコンピューティング.....	9
2-1.	Echo State Network (ESN).....	9
2-2.	ESN モデルの設計.....	12
2-3.	複数リザーバーモデル(groupedESN).....	14
2-4.	オフライン学習（リッジ回帰）.....	15
2-5.	オンライン学習（RLS アルゴリズム）.....	16
2-6.	モデル性能評価.....	19
第3章	ハイパーパラメータ探索.....	20
3-1.	グリッドサーチとランダムサーチ.....	21
3-2.	ベイズ最適化.....	22
第4章	実験.....	23
4-1.	実験体系.....	23
4-2.	計算環境.....	23
4-3.	実験データ.....	23
第5章	結果および考察.....	26
5-1.	推進性能の通常予測とサイクル予測.....	26
5-2.	リザーバーのサイズ(Nx)の最適化.....	29
5-3.	ベイズ最適化の結果.....	33
5-4.	探索するデータ区間.....	37
5-5.	Long Short-Term Memory (LSTM)モデル.....	39
5-6.	放電電圧 V_d が変化するときの予測結果.....	40
5-7.	推進剤流量 m が変化するときの予測結果.....	45
5-8.	simpleESN と cycle-groupedESN の比較考察.....	49
第6章	結論.....	50
参考文献		52
謝辞		56
付録		57
A.	各種パラメータの導出.....	57
B.	過学習.....	58

C.	W の要素を決定する分布と予測性能の関係	59
D.	Cycle-simpleESN	60
E.	入力スケールリング a_{in} と非線形性の関係	61
F.	目標出力 V_{cg} のときのノード数 N_x 最適化	62
G.	ハイパーパラメータの分布	64
H.	データセット C	66
I.	プルーム画像による放電電流予測.....	70
J.	モンテカルロドロップアウトによる不確実性評価.....	72
K.	忘却係数 λ の最適化	73
L.	出力結合重み W_{out} の時間変化	75

第1章 序論

1-1. 研究背景

電気推進のホールスラストは、空間電荷制限則の影響を受けないため高密度のイオンを加速することができ、高い推力密度を実現する。さらには、電源の少ないコンパクトな設計ができるため高推力電力比 (60mW/kW) である[1,2]。これらの特徴から、ホールスラストは人工衛星市場の拡大とともに、衛星の軌道修正や維持のための主推進として利用されることが増えてきた[3]。

宇宙空間でのホールスラストの実用では、高推力や高比推力等の向上の他に、消費電力の抑制や長寿命化が求められる。これらを実現するためには、ホールスラストの最適作動条件下での電源制御が必要不可欠である。

最適作動条件の探索および電源制御は、地上からの通信で行っている。この作業による人的コストは、衛星市場の拡大とともに無視できなくなると考えられる。人的コストの削減のためには、この電源制御をホールスラスト自身で完結させる自律型電源処理ユニット (Stand-Alone Power Processing Unit, SA-PPU) の開発が求められる。

1-2. ホールスラスト

1-2-1 作動原理および基本設計

ホールスラストの作動原理を図 1-1 で示す。ホールスラストは円環状のプラズマ加速チャネルとアノード、電子源の役割を担うカソード、コイルから構成されている。コイルは中心軸上と外周上に配置されており半径方向に磁場を印加している。カソードは電子を放出し、チャネル内に供給される。この電子は磁場によってトラップされ $\mathbf{E} \times \mathbf{B}$ ドリフトによってチャネル内に閉じ込められる。推進剤が加速チャネルの上流側から供給され、チャネル出口付近でトラップされている電子と衝突する。衝突すると陽イオンが生成され、軸方向に印加されている電場によって加速し放出される。この加速の力の反作用を用いてホールスラストは推力を得る。イオンのみが $\mathbf{E} \times \mathbf{B}$ ドリフトにトラップされずに加速できるのは、加速チャネルの長さを電子のサイクロトロン半径よりも長く、かつ、イオンのサイクロトロン半径よりも短く設計するからである。

カソードは中和器の役割も果たしており、排出されたイオンがカソード排出の電子によって中和され、排出イオンがスラストに引き戻されることを防ぐ。加速チャネル領域は常に電子が存在するため、電気的に準中性が保たれる。これらの仕組みから、空間電荷制限則を受けないため、高い推力密度を維持することが可能になる。

ホールスラストは以下の条件式(1.1)~(1.3)を満たすように設計する[4]。これにより、電子のみが周方向にトラップされ、 $\mathbf{E} \times \mathbf{B}$ ドリフトしてホール電流を発生させ、

イオンは印加した電場によって軸方向に加速する．

$$\omega_e \tau_e \gg 1 \quad (1.1)$$

$$r_e \ll L \ll r_i \quad (1.2)$$

$$L \ll \lambda_m \quad (1.3)$$

ここで ω_e [Hz]は電子のサイクロトロン周波数、 τ_e [sec]は電子の衝突時間、 r_e [m]と r_i [m]はそれぞれ電子とイオンのラーマー半径、 L [m]は加速チャンネル長、 λ_m [m]はイオンの平均自由行程である．式(1.1)はチャンネル内部で電子が $\mathbf{E} \times \mathbf{B}$ ドリフトをする条件である． $\omega_e \tau_e$ は電子が衝突間で旋回運動する回数を意味し、ホールスラストの場合は200~300程度だという報告がある[5]．式(1.2)は電子のみがトラップされ、イオンは磁場の影響を無視してチャンネル外に排出される条件である．式(1.3)は生成したイオンが再結合しないための条件である．各パラメータの詳細な導出は付録Aに記載している．

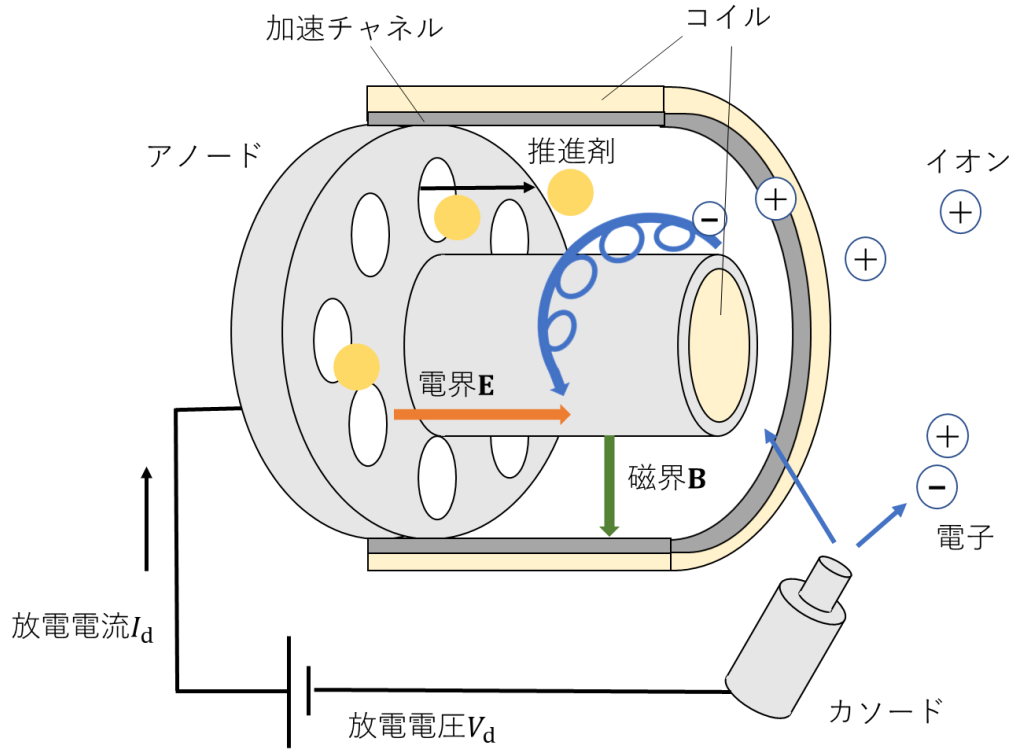


図 1-1 ホールスラストの作動原理[4].

1-2-2 放電電流の非線形性

ホールスラストの消費電力 P は放電電圧 V_d と放電電流 I_d を用いて、

$$P = V_d I_d \quad (1.4)$$

と表現できる．放電電圧 V_d の値は，電源制御により決めることができる．一方で，放電電流 I_d は，放電電圧 V_d 等の作動条件に応答する計測値である．したがって，放電電圧 V_d のように予め放電電流 I_d を決定することが出来ない．放電電圧 V_d 一定の条件下においては，放電電流 I_d が小さい条件で作動させることで消費電力の抑制につながる．最適作動条件を得るためにコイル電流を図 1-2 のように振り磁場強度に変化をさせて放電電流 I_d の応答を調べる．

放電電流 I_d は非線形な時間変化をするパラメータである．図 1-2 (a)に 200 W 級ホールスラストのコイル電流 I_d と放電電流 I_d の応答を図示する[6]．推進剤流量は 0.68mg/s，放電電圧 V_d は 350 V に設定し，コイル電流は線形的に増加(UP)と減少(DOWN)するように調節している．放電電流は非線形な応答を示している．さらには，プラズマが磁場に対してヒステリシス(履歴効果)を持つ影響で，増加時と減少時で放電電流値が異なる．同様に，図 1-2 (b)に同じ 200 W ホールスラストの別の作動実験の結果を示す．推進剤流量 0.68 mg/s，放電電圧 150 V に固定し，コイル電流の増加と減少時の放電電流応答を示している．コイル電流が 0.55~0.60A 付近のときに，放電電流が急激に増加もしくは減少しプロットが飛んでいる．

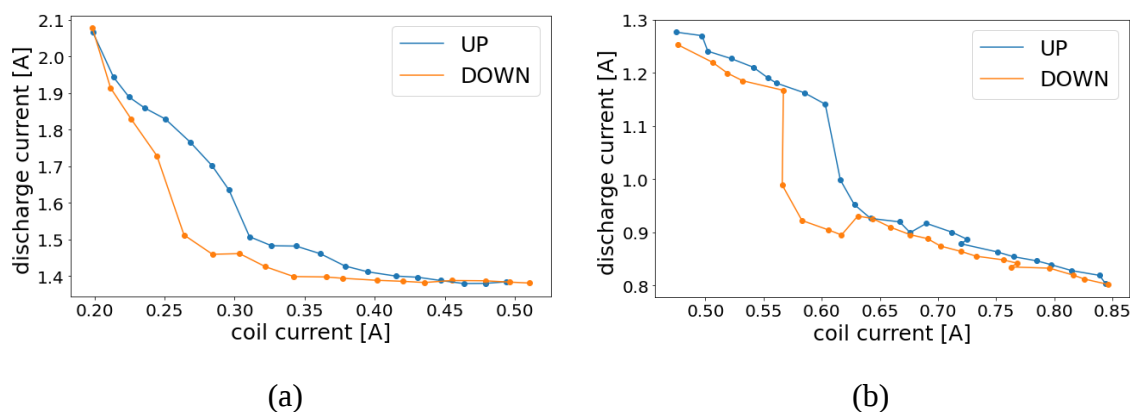


図 1-2 放電電流の非線形性

1-2-3 ホールスラスタの寿命

ホールスラスタの加速チャンネル壁面は、イオン衝撃スパッタリングによって浸食が発生し、チャンネル形状が変化してしまう。図 1-3 に作動初期と 5730 時間作動させたときのホールスラスタのチャンネル形状を示している[7]。この損耗によるチャンネル形状の変化がスラスタの性能劣化を招き、ホールスラスタの寿命低下につながる。磁気シールドを採用することで大幅に損耗を低減することに成功した例も報告されており[8-11]、チャンネル壁面の損耗が耐用寿命決定の主原因からは外れつつある。また、損耗評価モデルの開発も盛んに行われており、壁面浸食率などの評価が行われている[12-15]。一方で、損耗率ではなく推進性能の経年劣化度合いを上手くモデル化し予測している研究は少ない。その理由は、損耗の進行度合いが作動条件の履歴、放電チャンネルの材質やサイズ等に依存するからである。これらが複雑に絡み合っ、損耗度合いが決まるため、各スラスタに対して固有かつ宇宙実証中にモデルの改良が可能な枠組みが必要になる。

もう 1 つの寿命決定因子は、カソードの損耗である。ホールスラスタの耐久試験中に、カソードのスパッタリング現象による急激な損耗の発生が報告されている[16,17]。ここで、接地電位（宇宙空間では衛星）に対するカソード電位を V_{cg} と定義する。また、接地電位とイオンビームの浮遊電位間の電位差をプラズマ電位 V_p と定義する。電位差 $V_p + |V_{cg}|$ を利用して電子を放出する。一方で、イオンもカソード方向に逆流している。 $|V_{cg}|$ が大きくなると、逆流するイオンも増えてしまいスパッタリングが発生し損耗する。 V_{cg} はカソードの位置や角度、推進剤流量等の作動条件に依存するため、予め見積もることが難しい[18,19]。カソードの損耗を抑制するには、 $|V_{cg}|$ が小さくなる最適な作動条件を見つけ出し、そのパラメータで制御することが必要になる。カソードが故障すると電子の供給が停止するため、ホールスラスタの作動を止めなければならない。したがって、カソード損耗を抑制した最適作動条件での運転はホールスラスタの長寿命化に繋がる。

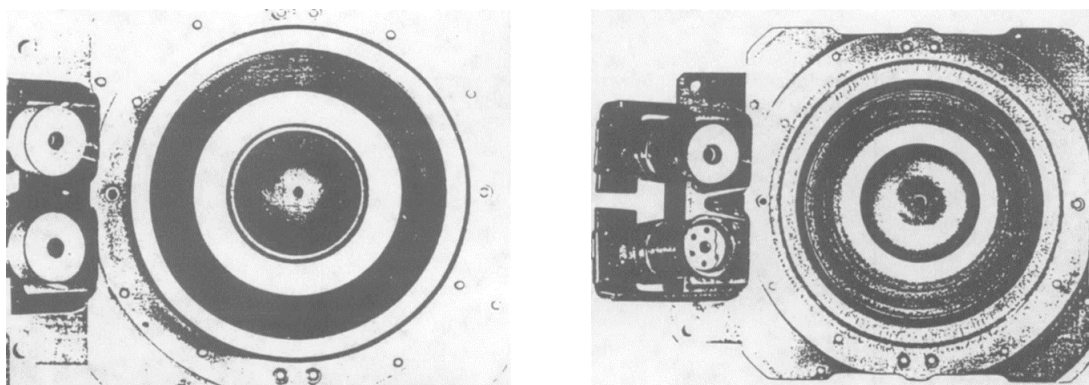


図 1-3 作動初期と 5730 時間作動後のホールスラスタ[7]

1-3. 研究目的

本研究の最終的な目標は、ホールスラスタに直接コンピューターを搭載して、オンボードで制御を行う自律型電源処理ユニット (SA-PPU) の開発である。近年では、松永ら[20]が空間計算量を抑えながら PPU が放電電流振動、電源、放電電流を制御する機能を備えたアルゴリズムの開発に成功している。本研究の SA-PPU の機能としては、消費電力とカソードの損耗を低減する作動条件を予測し、その最適作動条件で電源制御をすることを想定する。この機能の実現には、作動条件を変えたときの推進性能の応答を予測する必要が求められる。本研究での推進性能は消費電力に関わる放電電流 I_d とカソードの損耗に関わる接地電位に対するカソード電位 V_{cg} の 2 パラメータを指す。推進性能の応答が予測できれば、逆算して最適作動条件を決定することができる。

この推進性能の応答は数式を用いてモデリングすることが難しい[21]。その理由として、推進性能が非線形性を有している点(1-2-2 節)が挙げられる。また、ホールスラスタの性能はカソードの種類や配置位置や推進剤の種類、作動条件や経年変化の度合い(1-2-3 節)が複雑に絡み合っているため、汎用的なモデル構築は現実的ではない。作動条件(放電電圧、推進剤流量、コイル電流)に対する放電電流の応答をマップ化する取り組みも行われている[22]。しかし、計算資源が限られる PPU のメモリでは、マッピングした情報をすべて記憶させることは難しい[20]。

そこで本研究では「ニューラルネットワーク」手法を採用して推進性能の応答を予測する。ニューラルネットワークは既存データの学習を通して、入力データ(作動条件)と出力データ(推進性能)の関係を記述することができる。この手法を使うことの強みは、ニューラルネットワークの活性化関数により非線形なデータの予測も出来る点である。また、実験データさえあればモデルが構築できるのでホールスラスタ各々に適応したモデルを構築できる点も強みである。これらの強みは、前述の数式によるモデリングの課題を解消する。

先行研究でも、データ駆動型手法のニューラルネットワークを採用することで、推進性能を予測することに成功している[23,24]。順伝播型ニューラルネットワークと畳み込みニューラルネットワーク (CNN) の融合モデルを用いて作動条件とプルーム画像の入力情報から放電電流値を絶対予測誤差平均 0.8 %で予測できた。しかしながら、このモデルの学習には 5 時間程度 (CPU: Intel® Core™ i5-4590 で計算)が必要であった。学習時間が長いのは、計算資源が限られるオンボードコンピューティングでは致命的である。さらには、作動条件や損耗の変化によって推進性能も変化してしまうために、地上試験のみや宇宙実証の初期のデータのみを用いて学習させるわけにもいかず、作動中も学習を行ってモデルを更新する必要がある。以上から、学習時間を抑えたニューラルネットワークモデルが必要である。

本研究では学習時間が小さいとされるリザバーコンピューティングを用いて、低学習時間かつ高精度に推進性能の放電電流 I_d と接地電位に対するカソード電位 V_{cg}

を予測するモデルを構築することを目的とした。その際、1-2-3 節で述べた経年劣化による推進性能の変化にも対応できるように、リザーバーコンピューティングモデルを更新しづけるオンライン学習手法を適用した。リザーバーコンピューティングのモデル 2 種類を用意し比較を行っている。また、モデルの構造を最適化するために、ベイズ最適化を用いてハイパーパラメータの探索を行った。ここで、ハイパーパラメータとは、学習前に予め設定しておくモデルのパラメータのことを指す。

第2章 リザーバーコンピューティング

この節ではリザーバーコンピューティングの代表的なモデル，Echo State Network(ESN)モデルについて主に解説する。

2-1. Echo State Network (ESN)

Echo State Network モデルは Herbert Jaeger によって提唱された，結合重みを固定したリカレントニューラルネットワーク (RNN) である[25]。

まず，リカレントニューラルネットワーク (RNN) について説明する．RNN の特徴は，ネットワーク内部にフィードバック結合をもっていることであり．このフィードバック結合のおかげで，過去の情報を保持しながら学習を行うことができる．図 2-1 で RNN の学習のイメージを示す．この特徴から時系列データを学習データとして使うことが出来る．RNN を用いた応用例としては，モデル予測制御[26]，異常検知[27]，残存耐用時間予測[28]などが挙げられ，自然言語処理の分野以外でも幅広く研究されていることがわかる．RNN の代表的なモデルが Long Short-Term Memory (LSTM)[29]である．このモデルはゲート構造と呼ばれる特殊な構造をもつことで，長期的な情報の保持をしながら学習を進めることが出来る．しかしながら，LSTM は複雑な内部構造を訓練するために多くの学習時間とメモリを必要とする．

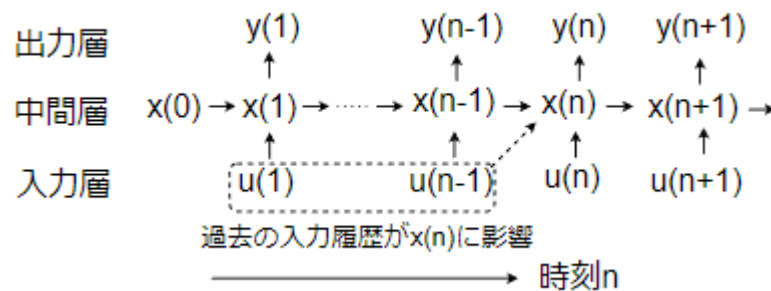


図 2-1 RNN の学習[30]

ESN の概略図を図 2-2 に示す。ESN は入力層、リザーバー層、出力層の 3 つの層から構成されている。通常の RNN では勾配法を用いて出力誤差を最小化するように結合重み（図の赤矢印と青矢印）を逐次更新して学習を完了させる。一方で、ESN は赤矢印で示した結合重み、入力とリザーバー間の入力結合重み W^{in} 、リザーバー同士のリカレント結合重み W を学習前に固定しておき、リザーバーと出力間の出力結合重み $W^{\text{out}}(n)$ のみを学習させる。このとき学習は勾配法ではなく、線形学習器をもちいる。線形学習器としては、4-5 節、4-6 節で紹介するリッジ回帰や逐次最小二乗法 (RLS 法)[31,32] などがある。線形学習器の計算量は、勾配法に基づく学習アルゴリズムの計算量に比べて、多くの場合圧倒的に小さい[30]。調節する結合重みを限定し、線形学習器を学習アルゴリズムとして採用できることから、ESN は LSTM などの他の RNN に比べて計算時間が小さい事が予想される。

ESN の入力 $\mathbf{u}(n)$ が与えられてからモデル出力 $\mathbf{y}(n)$ を得るまでの過程を説明する。まず、タイムステップ n での入力データ $\mathbf{u}(n)$ 、リザーバーの状態ベクトル $\mathbf{x}(n)$ 、予測される実験データ(以後、目標出力) $\mathbf{d}(n)$ をそれぞれ、

$$\mathbf{u}(n) = (u_1(n), \dots, u_{N_u}(n))^T \in \mathbb{R}^{N_u} \quad (2.1)$$

$$\mathbf{x}(n) = (x_1(n), \dots, x_{N_x}(n))^T \in \mathbb{R}^{N_x} \quad (2.2)$$

$$\mathbf{d}(n) = (d_1(n), \dots, d_{N_d}(n))^T \in \mathbb{R}^{N_d} \quad (2.3)$$

と表記する。ここで、 N_u は入力データの次元 (2 つの入力パラメータの場合、 $N_u = 2$)、 N_x はリザーバーのノード数、 N_d は目標出力の次元(予測するパラメータが 2 つの場合、 $N_d = 2$)である。リザーバーの状態ベクトル $\mathbf{x}(n)$ は式(2.4)によって得られる。

$$\mathbf{x}(n) = \tanh(W\mathbf{x}(n-1) + W^{\text{in}}\mathbf{u}(n)) \quad (2.4)$$

式(2.4)の右辺の第 1 項に $\mathbf{x}(n-1)$ が含まれる。これにより、過去の情報を引き継いだ状態で学習を進めることが出来る。入力結合重み行列 $W^{\text{in}} \in \mathbb{R}^{N_x \times N_u}$ 、リザーバーのリカレント結合重み行列 $W \in \mathbb{R}^{N_x \times N_x}$ は学習前に固定される行列である。また、 $\tanh(-)$ は活性化関数と呼ばれるものである。

活性化関数 $f(h)$ は内部状態 h を線形または非線形変換する。これは、人間の神経細胞（ニューロン）が発火する様子を模している。今回用いた双極正接 ($\tanh$) 関数は、図 2-3 で示すように、 $[-1, 1]$ の連続値をとる。内部状態 h の絶対値が大きい場合は非線性の強い変換が行われ、大きすぎると -1 または 1 に収束する。一方で、内部状態 h が 0 に近い場合は、線形変換となる[33]。

最終的に出力されるモデル出力 $\mathbf{y}(n)$ は、式(2.4)とタイムステップ n の出力結合重み行列 $W^{\text{out}}(n)$ を用いて、次のように表す。

$$\mathbf{y}(n) = W^{\text{out}}(n)\mathbf{x}(n) \quad (2.5)$$

このモデル出力 $\mathbf{y}(n)$ と目標出力 $\mathbf{d}(n)$ の誤差 $e(n)$ を最小化するように、線形学習器を

用いて出力結合重み行列 $W^{\text{out}}(n)$ を調節する. このとき, 出力結合重み行列 $W^{\text{out}}(n)$ は

$$W^{\text{out}}(n) = \left(w_1^{\text{out}}(n), \dots, w_{N_d}^{\text{out}}(n) \right)^{\top} \in \mathbb{R}^{N_d \times N_x} \quad (2.6)$$

で構成される.

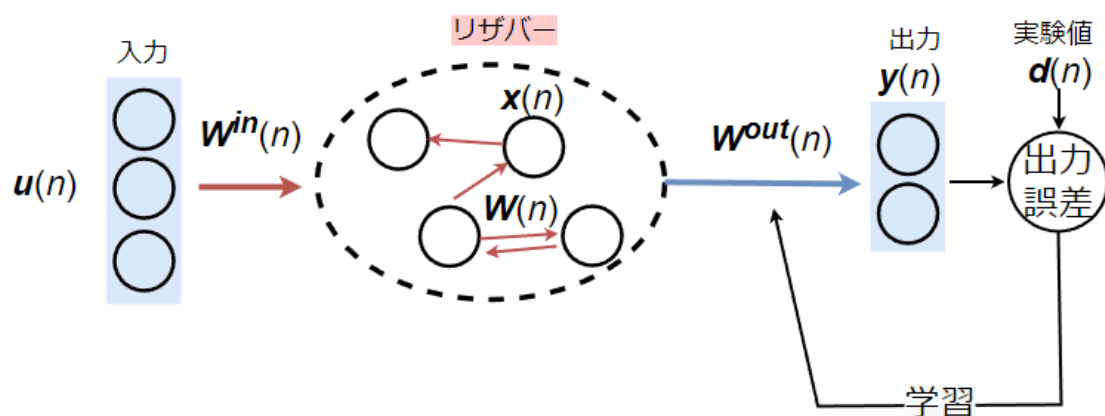


図 2-2 ESN の概略図

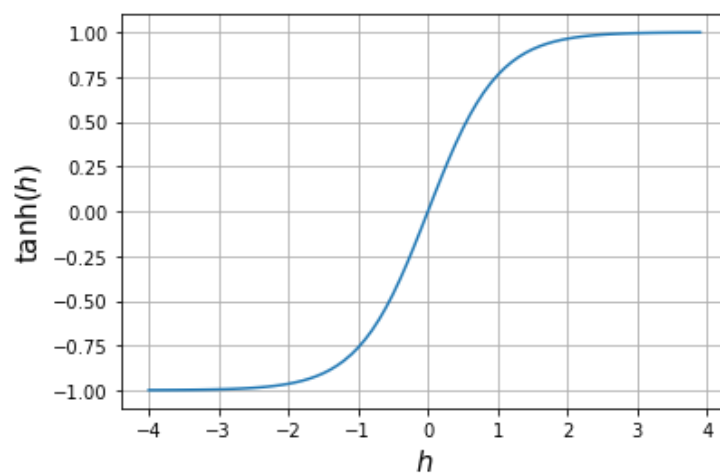


図 2-3 双極正接(tanh)関数

2-2. ESN モデルの設計

ESN モデルは学習前に固定するパラメータ（ハイパーパラメータ）の値を適切に設定することで、予測性能の良いモデルが構築される。パラメータの決定が、そのまま予測精度の良し悪しに直結する可能性もある。以下に示すハイパーパラメータはモデルの複雑さ、表現能力、予測性能に影響を与える[30].

[1] 入力結合重み行列 W^{in}

W^{in} は式(2.4)で用いられており、入力データをリザーバーに渡すときの結合重みである。この行列は、一様分布に従う乱数を与える。そのとき、分布の範囲は $[-a_{\text{in}}, a_{\text{in}}]$ とし、 a_{in} を入力スケーリングと名付ける。本研究ではこの入力スケーリングが、 W^{in} のハイパーパラメータであり、第3章で記載するハイパーパラメータ探索によって最適化される。

[2] リザーバーのリカレント結合重み W

リカレント結合重み W を設定するためには、リザーバーのサイズ、スパース(疎)性、スケーリングを決める必要がある。

リザーバーのサイズは、ノード数 N_x で表現される。式(2.6)より、この N_x は ESN の学習パラメータ数と比例するため適切に決める必要がある。 N_x が小さいと、モデルの表現力が不十分になる。一方で大きすぎると、モデルが複雑で表現力が高くなりすぎてしまい過学習(付録 B)と呼ばれる現象を引き起こす可能性がある。過学習がおこると、未学習のテストデータに対しての予測精度が悪くなる。また、4-5 節、4-6 節で紹介する線形学習器の時間計算量は $O(N_x^2 T)$ となるため、大きい N_x の選択は、計算コストの増大につながる[34]。ここで、 T は学習させるタイムステップ数である。本研究では、適切な N_x を見つけるために、 N_x と予測誤差の関係を示したマップを作成し、モデル設計の指針とした。

次に、リザーバーのスパース(疎)性について述べる。スパースなネットワークを用いることで、ノード状態の均一化を防ぎ入力に対する多様な応答を各ノードで得ることができる[30]。 $W \in \mathbb{R}^{N_x \times N_x}$ の各要素は確率 p で非ゼロ要素とし、確率 $1-p$ で0とする。この確率 p をハイパーパラメータとして最適化を行い、スパース度合いを決定する。

次に、 W の非ゼロ要素の設定方法について述べる。入力結合重み行列 W^{in} と同様に、特定の分布に従った乱数を与える。分布としては、区間 $[-1, 1]$ の一様分布、平均0、分散1の正規分布、 $\{-1, 1\}$ からランダムに選択する二値分布のいずれかを選択する。事前の調査で、この3種類の分布を用いて、推進性能の予測を行ったが、予測精度に顕著な差は見られなかった(付録 C)。従って、これ以降の ESN の設計では、区間 $[-1, 1]$ の一様分布を採用している。

ここで、一様分布の区間は予め $[-1, 1]$ と決められているため、適切なスケーリングを施す必要がある。このとき、スペクトル半径 ρ というものをを用いる。 $\rho(W)$ は W

の固有値の絶対値の最大値を意味する．スケーリング後のスペクトル半径 $\rho(W)$ が、

$$\rho(W) < 1 \quad (2.7)$$

を満たすようにスケーリングを行う． ESN モデルの設計の際には、 $\rho(W) < 1$ となるハイパーパラメータ $\rho(W)$ を予め指定しておく．本研究では、この $\rho(W)$ をハイパーパラメータ探索によって最適化する．

式(2.7)の条件は ESN の特徴である Echo State Property (ESP)[33]が成立するための必要条件である． ESP は初期依存性を排除するための性質である． 式(2.2)の初期値 $\mathbf{x}(0) \in \mathbb{R}^{N_x}$ は乱数シードを固定しない場合はランダムに与えられる． 式(2.4)を帰納的に変換していくと、 $\mathbf{x}(n)$ は初期値 $\mathbf{x}(0)$ の影響を受けることがわかる． この影響を無視できないと、ランダムな初期値 $\mathbf{x}(0)$ に応じて式(2.9)で示す出力が変化してしまう． 従って、時間発展とともに初期値 $\mathbf{x}(0)$ に依存しない $\mathbf{x}(n)$ を得る必要がある．

k

2-3. 複数リザーバーモデル(groupedESN)

本研究では、前述した ESN モデルのリザーバーを複数個用意するモデルを作成した。リザーバーがひとつのスタンダードな ESN モデルを以後、simpleESN と呼び、複数リザーバーの ESN モデルを groupedESN[35]と名付ける。図 2-4 に、リザーバーが2つのときの groupedESN モデルの概略図を示す。入力パラメータを u_1 と u_2 の2種類に分類しそれぞれを異なるリザーバーに入力する。それぞれのリザーバーで独立してハイパーパラメータを調節する。独立した設計が可能なので、それぞれの入力パラメータに合った、ノード数 N_x や結合重みのスケーリングを決めることができる。式(2.5)を計算する前に、各リザーバーで得られたリザーバー状態ベクトル $x^{\text{reservoir } i}(n)$ ($i = 1, 2, \dots, m$)は、以下に示すように結合しておく。

$$x^{\text{multi}}(n) = \begin{pmatrix} x^{\text{reservoir } 1}(n) \\ \vdots \\ x^{\text{reservoir } i}(n) \\ \vdots \\ x^{\text{reservoir } m}(n) \end{pmatrix} \quad (2.8)$$

groupedESN では、 $x^{\text{multi}}(n)$ を式(2.5)の $x(n)$ として扱う。

リザーバーコンピューティング分野では、複数リザーバーの ESN モデルの研究がさかんに行われている[35,36]，マルチタイムスケールの時系列データ群の予測タスクで高い予測精度を得たという報告もある[37]。

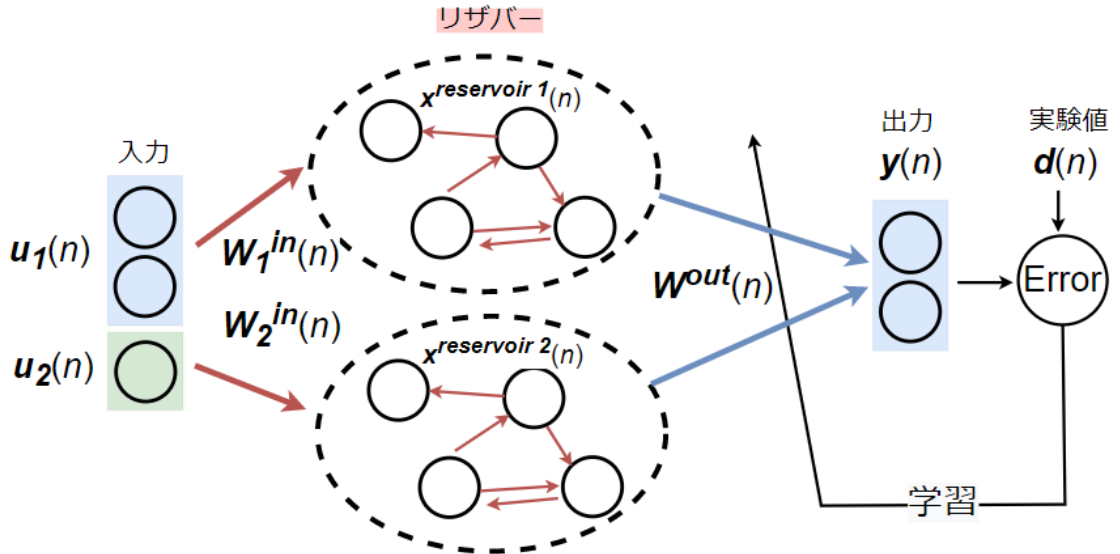


図 2-4 groupedESN の概略図

2-4. オフライン学習（リッジ回帰）

ESN モデルの学習アルゴリズムは線形学習器が用いられ、出力結合重み行列 $W^{\text{out}}(n)$ を最適化する。まず、この節でオフライン学習を行うリッジ回帰について説明する。オフライン学習はまとまったデータを一括で学習を行い、一回の学習で $W^{\text{out}}(n)$ の更新を一度のみ行う。したがって、学習済みモデルで予測タスクを実行している最中にデータの特性が変化してしまった場合には、予測精度が悪くなってしまう。これを解決するためには、データの貯蔵と学習を定期的に繰り返すか次節で述べるオンライン学習によって防ぐことができる。

まず、線形回帰から説明する。全タイムステップの目標出力 D とモデル出力 Y を以下のように表現する。

$$D = (d(1), d(2), \dots, d(n))^T \in \mathbb{R}^{N_d \times T} \quad (2.9)$$

$$Y = (y(1), y(2), \dots, y(n))^T \in \mathbb{R}^{N_y \times T} \quad (2.10)$$

ここで、 N_y は予測パラメータの次元、 T は学習データの総タイムステップ数である。同様に、全タイムステップのリザーバー状態ベクトルを結合した行列 $X(n)$ を

$$X(n) = (x(1), x(2), \dots, x(n)) \in \mathbb{R}^{N_x \times T} \quad (2.11)$$

と定義する。タイムステップ n までの出力誤差 $e(n)$ は、

$$e(n) = (e(1), e(2), \dots, e(n)) \quad (2.12)$$

と表現する。式(2.9)(2.10)(2.11)(2.12)から $W^{\text{out}}(n)$ を用いたタイムステップ n までの出力誤差

$$e(n) = D - W^{\text{out}}(n)X(n) \quad (2.13)$$

を得る。線形回帰のコスト関数は $e(n)$ の二乗誤差の総和 E であり、

$$E = e^T(n)e(n) \quad (2.14)$$

を最小化するように、線形学習器は $W^{\text{out}}(n)$ を最適化する。

リッジ回帰[38]は式(2.14)に正則化項と呼ばれるものを追加する。リッジ回帰のコスト関数は、

$$E_{\text{ridge}} = e^T(n)e(n) + \beta W^{\text{out}^T}(n)W^{\text{out}}(n) \quad (2.15)$$

とする。式(2.15)は第二項に正則化項(ペナルティ項)が追加されている。学習パラメータ（今回の場合は $W^{\text{out}}(n)$ ）の二乗値が大きい場合、モデルの表現力が高くなり過ぎてしまい過学習を起してしまう可能性がある。これを抑制するために第二項が追加される。ここで、 $\beta > 0$ は正則化パラメータであり、正則化項の影響度合いを決定する。

式(2.15)を最小化する $W^{\text{out}}(n)$ を求めるために、 $\partial E_{\text{ridge}} / \partial W^{\text{out}}(n) = 0$ の方程式を解く。以下に、この方程式を解く過程を示す。式(2.13)(2.15)より

$$\frac{\partial}{\partial W^{\text{out}}(n)} (\mathbf{e}^T(n)\mathbf{e}(n) + \beta W^{\text{out}}(n)^2) = 0 \quad (2.16)$$

$$\begin{aligned} \frac{\partial}{\partial W^{\text{out}}(n)} (D^T D - 2D^T W^{\text{out}}(n)X(n) + \\ X^T(n)W^{\text{out}}(n)^T W^{\text{out}}(n)X(n) + \beta W^{\text{out}}(n)^2) = 0 \end{aligned} \quad (2.17)$$

$$-2DX^T(n) + 2W^{\text{out}}(n)X(n)X^T(n) + 2\beta W^{\text{out}}(n) = 0 \quad (2.18)$$

この方程式の解は、

$$W^{\text{out}}(n) = DX^T(n)(X(n)X^T(n) + \beta I)^{-1} \quad (2.19)$$

となる．ここで、 $I \in \mathbb{R}^{N_x \times N_x}$ は単位行列である．式(2.19)の解を一意に得ることが、リッジ回帰における学習の終了を意味する．

2-5. オンライン学習 (RLS アルゴリズム)

オンライン学習とは、まとまったデータを使って学習を行うオフライン学習とは異なり、データがひとつ与えられるたびに逐次的に学習パラメータを更新する手法である．この方法を用いることで、学習済みモデルでの予測時にデータの特徴が変化してしまう現象にも対応できる．損耗等によりホールスラストの推進性能が違うモードに遷移した場合も、即座に学習し予測することが可能になる．本節では、オンライン学習アルゴリズムの逐次最小二乗 (RLS) 法について述べる．ホールスラストの推進性能予測において、RLS 法は忘却係数 λ を用いることで過去のデータの影響を小さくしていきながら、学習パラメータを更新できる利点を持つ．これにより、ホールスラストの作動条件の変更や損耗等による推進性能の経年変化をうまく反映することが見込める．言い換えると、期間中の全データに対しての学習パラメータを決定せず、予測ステップの直近のデータの影響を強く反映したモデルの構築が可能になる．

以降で、ESN モデルの学習パラメータ、出力結合重み行列 $W^{\text{out}}(n)$ を更新する漸化式を導出する．導出は文献[30,32]を参考にしている．RLS 法は出力されるモデル出力の次元数が1と限定されているため、 $N_d=1$ の場合を考える．その上で最後にモデル出力の次元が複数の場合に対応させる．

まず、 $N_d=1$ のときの全タイムステップの目標出力を、

$$\mathbf{D}_1 = (d_1(1), d_1(2), \dots, d_1(n))^T \in \mathbb{R}^T \quad (2.20)$$

とする．式(2.6)(2.11)(2.12)(2.20)より、タイムステップ n の出力結合重み $\mathbf{w}_1^{\text{out}}(n)$ を用いた出力誤差は、

$$\mathbf{e}(n) = \mathbf{D}_1 - X^T \mathbf{w}_1^{\text{out}}(n) \quad (2.21)$$

と表現する．そして、RLS 法のコスト関数は

$$E_{\text{rls}} = \mathbf{e}(n)^T \mathbf{\Lambda}(n) \mathbf{e}(n) \quad (2.22)$$

となる．ここで、 $\mathbf{\Lambda}(n)$ は忘却係数 λ の対角行列であり、

$$\Lambda(n) = \begin{bmatrix} \lambda^{n-1} & 0 & \cdots & 0 \\ 0 & \lambda^{n-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad (2.23)$$

とする．忘却係数 λ は $0 < \lambda \leq 1$ と設定する．タイムステップの出力誤差が過去に遡るほど次数が大きくなり，結果的に古いタイムステップの誤差が E_{rls} に占める割合が小さくなる．これが RLS 法の直近のデータの影響を強く受けるという利点を実現する．

式(2.23)を最小化するために $\partial E_{\text{rls}} / \partial \mathbf{w}_1^{\text{out}}(n) = 0$ を解く．式(2.16)~(2.18)の方程式の導出を参考にしながら計算すると，

$$\hat{\mathbf{w}}_1^{\text{out}}(n) = \Phi_\lambda^{-1}(n) \Theta_\lambda(n) \quad (2.24)$$

を得る．ここで，

$$\Phi_\lambda(n) = X(n) \Lambda(n) X^\top(n) \quad (2.25)$$

$$\Theta_\lambda(n) = X(n) \Lambda(n) \mathbf{D}_1 \quad (2.26)$$

と置換している．式(2.11)(2.20)を(2.25)(2.26)に代入して展開すると，

$$\begin{aligned} \Phi_\lambda(n) &= \mathbf{x}(n) \mathbf{x}^\top(n) + \lambda \mathbf{x}(n-1) \mathbf{x}^\top(n-1) \\ &\quad + \lambda^2 \mathbf{x}(n-2) \mathbf{x}^\top(n-2) + \cdots \end{aligned} \quad (2.27)$$

$$\begin{aligned} \Theta_\lambda(n) &= \mathbf{x}(n) d_1(n) + \lambda \mathbf{x}(n-1) d_1(n-1) \\ &\quad + \lambda^2 \mathbf{x}(n-2) d_1(n-2) + \cdots \end{aligned} \quad (2.28)$$

を得る．式(2.27)と(2.28)は帰納的に，それぞれ

$$\Phi_\lambda(n) = \lambda \Phi_\lambda(n-1) + \mathbf{x}(n) \mathbf{x}^\top(n) \quad (2.29)$$

$$\Theta_\lambda(n) = \lambda \Theta_\lambda(n-1) + \mathbf{x}(n) d_1(n) \quad (2.30)$$

と変換できる．次に逆行列補題（逆行列の補助定理）[39]を用いて，式(2.29)の変換を行う．逆行列補題では，

$$(A + \alpha \mathbf{a} \mathbf{a}^\top)^{-1} = A - \frac{\alpha A^{-1} \mathbf{a} \mathbf{a}^\top A^{-1}}{1 + \alpha \mathbf{a}^\top A^{-1} \mathbf{a}} \quad (2.31)$$

が成立する．ここで， $A \in \mathbb{R}^{N \times N}$ は正方行列， $\mathbf{a} \in \mathbb{R}^N$ は列ベクトル， α は任意のスカラー値である．この性質を利用して式(2.29)は

$$\Phi_\lambda^{-1}(n) = \lambda^{-1} (\Phi_\lambda^{-1}(n-1) - \mathbf{g}(n) \mathbf{x}^\top(n) \Phi_\lambda^{-1}(n-1)) \quad (2.32)$$

と変換できる．ここで， $\mathbf{g}(n)$ はゲインベクトルであり

$$\mathbf{g}(n) = \frac{\lambda^{-1} \Phi_\lambda^{-1}(n-1) \mathbf{x}(n)}{1 + \lambda^{-1} \mathbf{x}^\top(n) \Phi_\lambda^{-1}(n-1) \mathbf{x}(n)} \quad (2.33)$$

と表す．式(2.33)を変換し式(2.32)を用いることで，

$$\begin{aligned} \mathbf{g}(n) &= \lambda^{-1} \Phi_\lambda^{-1}(n-1) - \mathbf{g}(n) \lambda^{-1} \mathbf{x}^\top(n) \Phi_\lambda^{-1}(n-1) \mathbf{x}(n) \\ &= \Phi_\lambda^{-1}(n) \mathbf{x}(n) \end{aligned} \quad (2.34)$$

と表現できる $\Phi_\lambda^{-1}(n)$ と $\mathbf{g}(n)$ の漸化式を得る．式(2.30)を式(2.24)に代入して，式(2.33)を代入すると，

$$\begin{aligned} \hat{\mathbf{w}}_1^{\text{out}}(n) &= \lambda \Phi_\lambda^{-1}(n) \Theta_\lambda(n-1) + \Phi_\lambda^{-1}(n) \mathbf{x}(n) d_1(n) \\ &= \lambda \Phi_\lambda^{-1}(n) \Theta_\lambda(n-1) + \mathbf{g}(n) d_1(n) \end{aligned} \quad (2.35)$$

を得る．式(2.24)(2.32)(2.34)を用いて，式(2.35)をさらに変換して，

$$\begin{aligned}
\hat{\mathbf{w}}_1^{\text{out}}(n) &= \Phi_\lambda^{-1}(n-1)\Theta_\lambda(n-1) - \mathbf{g}(n)\mathbf{x}^\top(n)\Phi_\lambda^{-1}(n-1)\Theta_\lambda(n-1) + \mathbf{g}(n)d_1 \\
&= \hat{\mathbf{w}}_1^{\text{out}}(n-1) + \mathbf{g}(n)(d_1 - \mathbf{x}^\top(n)\Phi_\lambda^{-1}(n-1)\Theta_\lambda(n-1)) \\
&= \hat{\mathbf{w}}_1^{\text{out}}(n-1) + \mathbf{g}(n)(d_1 - \mathbf{x}^\top(n)\hat{\mathbf{w}}_1^{\text{out}}(n-1)) \\
&= \hat{\mathbf{w}}_1^{\text{out}}(n-1) + \mathbf{g}(n)(d_1 - \hat{\mathbf{w}}_1^{\text{out}}(n-1)\mathbf{x}^\top(n))
\end{aligned} \tag{2.36}$$

を得る．このとき，更新前の出力結合重み行列を用いて，現タイムステップの出力誤差 $\hat{e}_1(n)$ は次のようにかける．

$$\hat{e}_1(n) = d_1 - \mathbf{w}_1^{\text{out}}(n-1)\mathbf{x}(n) \tag{2.37}$$

式(2.37)を式(2.36)に代入して，

$$\hat{\mathbf{w}}_1^{\text{out}}(n) = \hat{\mathbf{w}}_1^{\text{out}}(n-1) + \mathbf{g}(n)\hat{e}_1(n) \tag{2.38}$$

を得る．これで RLS 法に必要な漸化式が得られた．

次に，初期条件について考える．式(2.32)で $\Phi_\lambda(n)$ の逆行列 $\Phi_\lambda^{-1}(n)$ を生成している．したがって， $\Phi_\lambda(n)$ は逆行列の存在条件「正則行列かつ行列式が 0 でない」を満たす必要がある．そのために， $\Phi_\lambda(n)$ の初期条件は

$$\Phi_\lambda(0) = \delta I \tag{2.39}$$

とし， δ は正の実数の定数とする．この定数 δ はハイパーパラメータ探索によって最適化する．式(2.29)に式(2.39)を代入して，帰納的に計算すると

$$\Phi_\lambda(n) = X(n)\Lambda(n)X^\top(n) + \delta \lambda^n I \tag{2.40}$$

が得られた．式(2.25)はこの式(2.40)に置き換えられる．これは，式(2.22)の RLS 法のコスト関数を，

$$E_{\text{rls}} = \mathbf{e}(n)^\top \Lambda(n) \mathbf{e}(n) + \delta \lambda^n \mathbf{w}_1^{\text{out}\top}(n) \mathbf{w}_1^{\text{out}}(n) \tag{2.41}$$

と定義することと同値である．リッジ回帰と同様に，式(2.41)の第二項目が正則化項の役割を担う．また，出力結合重みベクトルの初期条件として

$$\hat{\mathbf{w}}_1^{\text{out}}(0) = \mathbf{0} \in \mathbb{R}^{N_x \times N_x} \tag{2.42}$$

とした．

式(2.38)が，RLS 法で学習パラメータを逐次的に更新するための漸化式である．式(2.32)の $\Phi_\lambda^{-1}(n)$ ，式(2.34)のゲインベクトル $\mathbf{g}(n)$ ，式(2.37)の更新前の結合重み $\hat{e}_1(n)$ を計算することで，式(2.38)を解くことができる．

以上より，モデル出力の次元数が 1 の場合の出力結合重みベクトル $\mathbf{w}_1^{\text{out}}(n)$ の更新則を得ることが出来た．本研究ではモデル出力の次元数が複数のものを扱うため，他の最適化された出力結合重みベクトル $\hat{\mathbf{w}}_{N_d}^{\text{out}}(n)$ も式(2.38)と同様にして得られる．

$$\hat{\mathbf{w}}_{N_d}^{\text{out}}(n) = \hat{\mathbf{w}}_{N_d}^{\text{out}}(n-1) + \mathbf{g}_{N_d}(n) \hat{e}_{N_d}(n) \tag{2.43}$$

このように，各出力パラメータに対して，漸化式(2.43)を解くことで，式(2.6)で表現される $\mathbf{W}^{\text{out}}(n)$ を得る．

2-6. モデル性能評価

予測精度の良し悪しを決定する誤差指標として，正規化平方平均二乗誤差（Normalized Root Mean Squared Error，以後 NRMSE）を用いた．NRMSE は次式で表す．

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{T} \sum_{t=k}^{k+T-1} \|\mathbf{d}(t) - \mathbf{y}(t)\|_2^2}}{\sqrt{\frac{1}{T} \sum_{t=k}^{k+T-1} \|\mathbf{d}(t) - \bar{\mathbf{d}}\|_2^2}} \quad (2.44)$$

ここで，分子は平方平均二乗誤差（Root Mean Squared Error，以後 RMSE）であり，モデル出力と目標出力間の誤差指標である． T は区間 $[k, k + T - 1]$ の総タイムステップ数である．RMSE は誤差を二乗しているため，絶対予測誤差平均に比べて外れ値の影響が大きくなる．この RMSE を目標出力の標準偏差で正規化することで NRMSE を求める． $\bar{\mathbf{d}} = \sum_{n=1}^T \mathbf{d}(n)/T$ は目標出力の時間方向の平均である．NRMSE が小さいほど，予測精度が高いことを示す．

第3章 ハイパーパラメータ探索

第2章では、ESNモデルを設計するために必要なパラメータが複数登場した。線形学習器によって最適化される出力結合重み行列 $W^{\text{out}}(n)$ 以外のパラメータは、ESNの学習前に、設定しておく必要のあるパラメータである。このパラメータのことをハイパーパラメータと呼ぶ。ハイパーパラメータの設定は、最初から固定した値を用いても構わないが、高精度に予測できるモデルを構築するにはハイパーパラメータ探索が必要である。ハイパーパラメータ探索の流れを以下に示す。

1. まず各パラメータの探索空間を設定する。
2. 探索空間内で、ハイパーパラメータの組み合わせを作成する
3. その組み合わせをモデルに設定して、モデルの性能評価を行う。
4. 1-3 を何度も繰り返すことで、一番性能がよかったハイパーパラメータを採用する。

ハイパーパラメータ探索には、手順2の組み合わせの決定方法を工夫して、いかに手順4の繰り返し回数を減らしながら、高性能のモデルを探してこられるかが重要となる。次節以降で、ハイパーパラメータ探索の基本的な手法（グリッドサーチとランダムサーチ）、ベイズ最適化手法について述べる。

3-1. グリッドサーチとランダムサーチ

グリッドサーチのイメージを図 3-1(a)に示す. 2 つのハイパーパラメータ α と β を仮定する. 各ハイパーパラメータは探索範囲で離散値をとるように設定される. グリッドサーチはその離散値の組み合わせをすべて試し, モデルの性能を評価する. グリッドサーチの利点は, すべての組み合わせを試すため, 一番性能の良いパラメータを取得できることである. しかしながら, ハイパーパラメータの数が多い場合や離散値間の幅を小さくしたい場合は, 組み合わせの数が膨大になり, 計算コストの増加につながってしまう[40].

図 3-1(b)に示すようにランダムサーチは, すべての組み合わせを試すのではなく, 任意の試行回数内でランダムに組み合わせを抽出し評価を行う. この方法は, 試行回数を決めることで計算コストを削減することが出来る. 一方で, 試行回数が小さい場合は最適な組み合わせを見つけられない確率が上がってしまう.

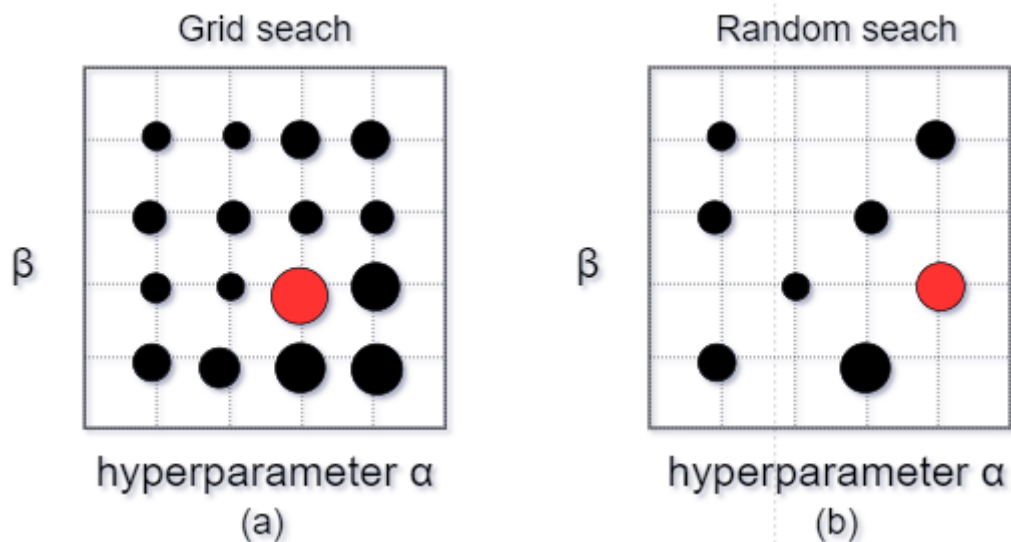


図 3-1 (a)グリッドサーチ (b)ランダムサーチのイメージ

3-2. ベイズ最適化

ベイズ最適化[41]はガウス過程回帰等を用いて、ハイパーパラメータの組み合わせと予測性能の関係を記述した獲得関数(Acquisition function)を推定する。その推定をもとに次に探索を行うハイパーパラメータの組み合わせを決定する。ベイズ最適化を用いた探索のイメージを図 3-2 に示す。ランダムサーチとは異なり、「探索と活用」がベイズ最適化の特徴である。「活用」は獲得関数により予測性能がよかった周辺領域の組み合わせを調べる。一方で、「探索」は局所解に陥るのを防ぐために情報の少ない領域の組み合わせを調べる。ベイズ最適化はグリッドサーチとランダムサーチと比較して、効率よく予測性能の高いハイパーパラメータの組み合わせを探し出すことが出来たことが報告されている[42]。

本研究では、獲得関数の構築に Tree-structured Parzen Estimator (TPE)[43]を採用して、ESN モデルのハイパーパラメータの最適化を行った。ベイズ最適化の実装には、オープンソースのハイパーパラメータ自動最適化フレームワーク Optuna(Preferred Networks 社)[44]を使用した。実際のコーディングにおいて、Optuna は最小化したい指標(NRMSE)などを関数の返り値に指定するだけで、ベイズ最適化によるハイパーパラメータ探索を行うことができる。並列処理の実装や探索結果の可視化にも優れるフレームワークである。

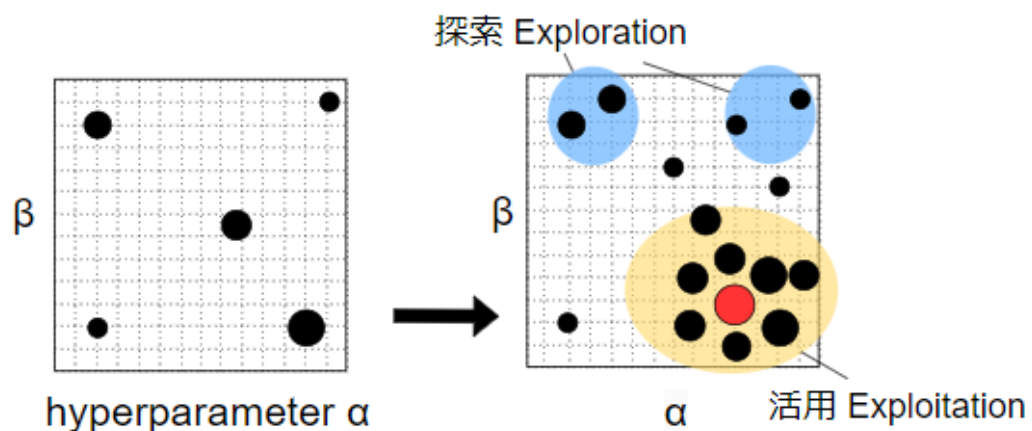


図 3-2 ベイズ最適化のイメージ

第4章 実験

4-1. 実験体系

実験データを取得するために、九州大学で開発した 200W 級のマグネティック型ホールスラスト[45]を用いた。加速チャネルは外径 56mm, 内径 40mm であり, 素材は窒化ホウ素である。ホールスラストの中心部に 1 つ, 外側に 4 つのソレノイドコイルを設置することで, 加速チャネル内半径方向に放射状の磁場を形成する[46]。コイル電流を変化させることで磁場密度を変えることを可能にする。電子源はホロカソード(HC252, Veeco 社)を使用し, ホールスラストの中心軸に対して 45 度方向に設置している。推進剤はキセノンを使用し, 流量 $\dot{m}$ [sccm]の制御を行った。

作動条件は, コイル電流 I_{coil} , 放電電圧 V_d [A], 推進剤流量 $\dot{m}$ [sccm]の 3 パラメータである。この 3 パラメータを変化させたときの推進性能を, ホールスラストの作動から取得する。推進性能は, 放電電流 I_d と接地電位に対するカソード電位 V_{cg} の 2 パラメータである。この 2 つのパラメータを作動条件のみで予測できるかを, ESN モデルを用いて検証する。

4-2. 計算環境

5 章以降での計算は, プロセッサ: Intel® Xeon® W-2235 CPU, 2.71, RAM: 96.0 GB, OS: Windows10 のデスクトップタイプの PC を使用した。Anaconda3 (Python3.8.5) を使用し, 必要なフレームワークは適宜 Anaconda 経由でインストールしている。インストールした主要なフレームワークは, ベイズ最適化用の optuna2.10.0 である。ベイズ最適化時に for 文を利用する場面では, 計算時間短縮のため Python パッケージの multiprocessing を使用して並列処理を行った。ただし, 並列処理による計算時間短縮に焦点を当てるのは本研究の趣旨から外れるので, 並列処理の有無による比較等を行っていない。以降で, 計算時間の記述がある場合は multiprocessing による並列処理は行っていないときの時間を記述することとする。

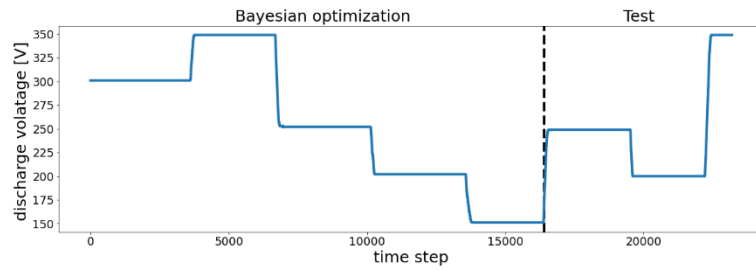
4-3. 実験データ

ホールスラストの作動実験から, 2 パターンのデータセットを取得した。それぞれのデータセットについて, 以下で図を用いながら述べる。

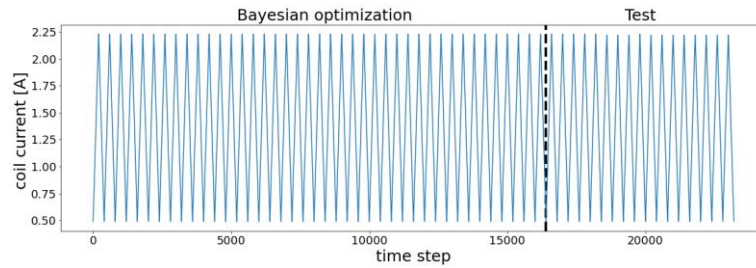
[データセット A]

ローパスフィルタされた信号をサンプリング周波数 1 kHz で取得し、ダウンサンプリングを行ってサンプリング周波数 40 Hz に変換している．放電電圧 V_d は図 4-1(a)に示すような変化をさせ，推進剤流量 $\dot{m}$ は 7 sccm (0.68 mg/s)で固定した．コイル電流 I_{coil} は図 4-1 (b)に示すように周波数 0.1 kHz で，0.5 A と 2.25 A の間を三角波振動している．この作動条件に応答する I_d を図 4-1 (c)，接地電位に対するカソード電位 V_{cg} を図 4-1(d)に示す．総タイムステップ数は 23200 ステップである．

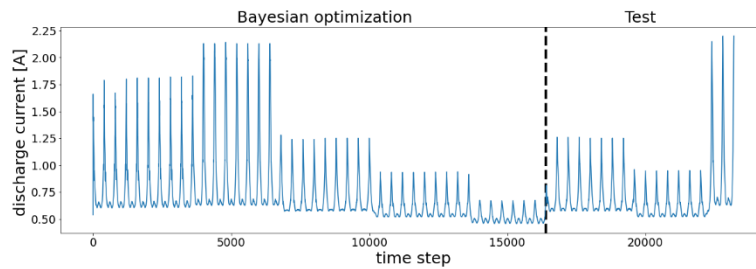
(a)



(b)



(c)



(d)

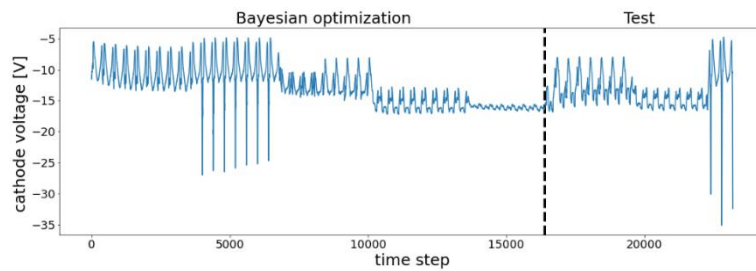
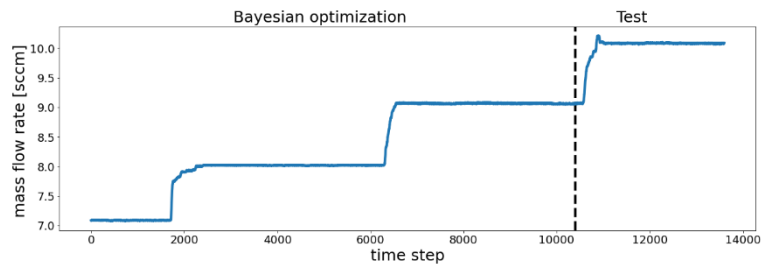


図 4-1 データセット A (a) V_d (b) I_{coil} (c) I_d (d) V_{cg}

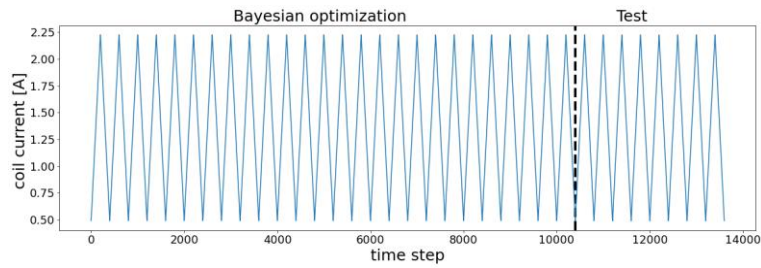
[データセット B]

ローパスフィルタされた信号をサンプリング周波数 2 kHz で取得し、ダウンサンプリングを行ってサンプリング周波数 40 Hz に加工している。 V_d を 300 V に固定して、 $\dot{m}$ は図 4-2(a) に示すような変化をさせる。コイル電流 I_{coil} は図 4-2 (b) で示すように周波数 0.1 Hz で、0.5 A と 2.25 A の間を三角波振動している。放電電流 I_d を図 4-2 (c), 接地電位に対するカソード電位 V_{cg} を図 4-2 (d) に示す。総タイムステップ数は 14000 ステップである。

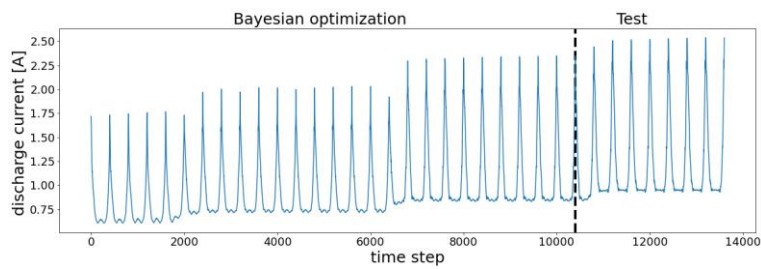
(a)



(b)



(c)



(d)

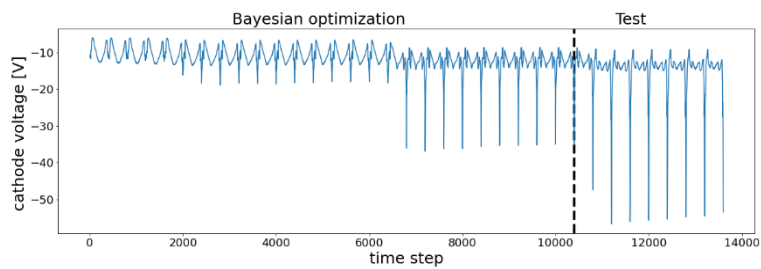


図 4-2 データセット B (a) $\dot{m}$ (b) I_{coil} (c) I_d (d) V_{cg}

第5章 結果および考察

5-1. 推進性能の通常予測とサイクル予測

ESN モデルの入力 $\mathbf{u}(n)$ に作動条件($V_d, \dot{m}, I_{\text{coil}}$), 目標出力 $\mathbf{d}(n)$ に I_d または V_{cg} を設定して, モデル出力 $\mathbf{y}(n)$ を得た. モデル出力 $\mathbf{y}(n)$ が推進性能の予測値である. データセット A, B は, ESN の入力 $\mathbf{u}(n)$ または目標出力 $\mathbf{d}(n)$ と使われる前に, 平均 0, 分散 1 の標準化を施している. ESN モデルの予測方法は 2 種類を用意した.

1 つ目は, 時刻 n の作動条件から時刻 n の推進性能を予測するスタンダードな方法である. 以後, この予測に用いる ESN モデルを simpleESN と名付ける. simpleESN を用いた予測のイメージを図 5-1 (a) に示す. また, 表 5-1 に simpleESN のハイパーパラメータと, 探索区間を示す. この探索空間は, 事前に文献[33]を参考にしながら, すべてのハイパーパラメータに対して, 1 探索の試行回数 512 のベイズ最適化を 512 探索して, 最適化されたハイパーパラメータの分布を得て決定している. この作業によって, 探索空間が見当はずれの領域になること, 探索空間が広がってしまうことを防いでいる.

2 つ目の予測方法は, コイル電流 I_{coil} の周期性を用いた方法である. データセット A とデータセット B のコイル電流は, 周波数 0.1 Hz で振動しており, 1 周期 400 ステップの離散データが存在する. この 1 サイクル分のデータの情報をすべて m サイクル目の入力 $\mathbf{u}(m)$ とし, m サイクル目の推進性能 I_d, V_{cg} を一度 400 ステップ分予測する. この予測モデルを cycle-groupedESN と名付け, 予測のイメージを図 5-1 (b) に示す. リザーバー 1 とリザーバー 2 の 2 つの独立したリザーバーを用意する. 放電電圧 V_d と推進剤流量 $\dot{m}$ は, 図 4-1.2 で示すように, 一定区間は一定値をとる. 従って, m サイクル目の入力は V_d と $\dot{m}$ の平均をそれぞれ計算して, リザーバー 1 への入力 $\mathbf{u}_1(m)$ として, 以下で表現する.

$$\mathbf{u}_1(m) = (\bar{V}_d(m), \bar{\dot{m}}(m))^T \in \mathbb{R}^2 \quad (5.1)$$

m サイクル目の 400 ステップ分のコイル電流を $I_{\text{coil}}(n), I_{\text{coil}}(n+1), \dots, I_{\text{coil}}(n+399)$ とした場合, リザーバー 2 への入力 $\mathbf{u}_2(m)$ にこの 400 ステップ分のコイル電流を入力する.

$$\mathbf{u}_2(m) = (I_{\text{coil}}(n), I_{\text{coil}}(n+1), \dots, I_{\text{coil}}(n+399))^T \in \mathbb{R}^{400} \quad (5.2)$$

$\mathbf{u}_1(m)$ は 2 次元に対して, $\mathbf{u}_2(m)$ は 400 次元である. $\mathbf{u}_1(m)$ と $\mathbf{u}_2(m)$ を 1 つのリザーバーに入力してしまうと, 次元数の少ない $\mathbf{u}_1(m)$ の情報を反映できず予測精度が悪くなると考えた. これを解消するために節 2-3 の groupedESN を採用し, $\mathbf{u}_1(m)$ と $\mathbf{u}_2(m)$ を別々のリザーバーに入力するモデルを構築した. 実際, $\mathbf{u}_1(m)$ と $\mathbf{u}_2(m)$ を同じリザーバーに入力したところ, N_x の値に大きく依存した不安定な予測を示す結果を得ている(付録 D). 式(2.4)と同様に, リザーバー 1 の状態ベクトルの時間発展 $\mathbf{x}_1(m)$,

リザーバー2 の状態ベクトルの時間発展 $\mathbf{x}_2(m)$ は次のように表現される．

$$\mathbf{x}_1(m) = \tanh\left(W_1\mathbf{x}_1(m-1) + W_1^{\text{in}}\mathbf{u}_1(m)\right) \quad (5.3)$$

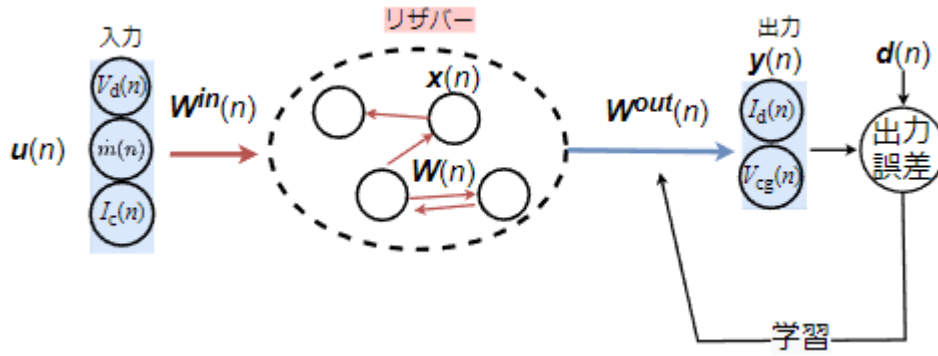
$$\mathbf{x}_2(m) = \tanh\left(W_2\mathbf{x}_2(m-1) + W_2^{\text{in}}\mathbf{u}_2(m)\right) \quad (5.4)$$

ここで， $\mathbf{x}_1(m) \in \mathbb{R}^{N_{x1}}$ と $\mathbf{x}_2(m) \in \mathbb{R}^{N_{x2}}$ であり， N_{x1} と N_{x2} は図中のリザーバー1 のノード数， リザーバー2 のノード数である． モデル出力 $\mathbf{y}(n)$ は式(2.5)(2.8)より

$$\mathbf{y}(m) = W^{\text{out}}(m) \begin{pmatrix} \mathbf{x}_1(m) \\ \mathbf{x}_2(m) \end{pmatrix} \quad (5.5)$$

として得る．

(a)



(b)

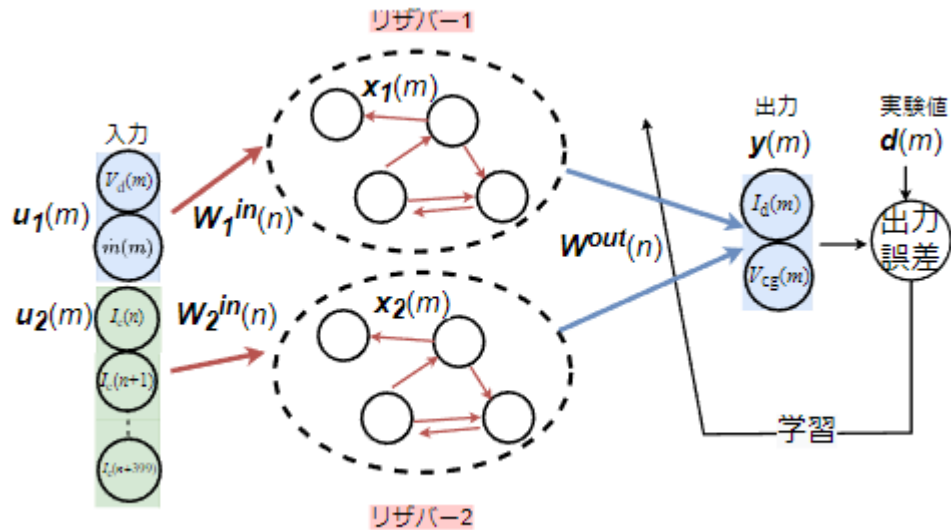


図 5-1 推進性能予測 ESN モデル (a) simpleESN (b) cycle-groupeESN

cycle-groupedESN の設計に必要なハイパーパラメータの一覧を，探索区間とともに表 5-2 に示す． $\mathbf{x}_1(m)$ と $\mathbf{x}_2(m)$ のそれぞれでハイパーパラメータ探索を行っている．このとき，探索空間は simpleESN のときと同様，あらかじめベイズ最適化を 512 探索行うことで決定している．表 5-1 と比べたときに，入力スケーリング a_{in} の探索区間だけ異なるものを用意している．これは，最適な入力スケーリング a_{in} が予測タスクの非線形性に依存するためである[33]．端的に述べると，cycle-groupedESN はデータの加工を施したことで予測タスクの非線形性が simpleESN の場合に比べて弱まったと考えられる． a_{in} と非線形性の関係についての詳細は付録 E に任せる．

表 5-1 simpleESN のハイパーパラメータ

ハイパーパラメータ	探索区間
入力スケーリング a_{in}	[1, 10]
リザーバーのノード数 N_x	[5, 200]
	50 以降は 10 刻み
結合確率 p	[0.01,0.5]
スペクトル半径 $\rho(W)$	[0.8, 0.99]
忘却係数 λ	[0.9, 0.9999]
正則化項の係数 δ	[0.00001, 0.001]

表 5-2 cycle-groupedESN のハイパーパラメータ

ハイパーパラメータ	探索区間
入力スケーリング a_{in1}, a_{in2}	[0.0001, 0.01]
リザーバーのノード数 N_{x1}, N_{x2}	[5, 200]
	50 以降は 10 刻み
結合確率 p_1, p_2	[0.01,0.5]
スペクトル半径 $\rho_1(W_1), \rho_2(W_2)$	[0.8, 0.99]
忘却係数 λ	[0.9, 0.9999]
正則化項の係数 δ	[0.00001, 0.001]

5-2. リザーバーのサイズ(N_x)の最適化

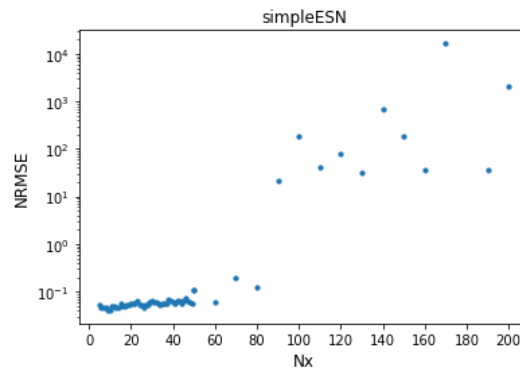
5-1 で提示したハイパーパラメータの中でも、リザーバーのノード数 N_x はモデルの複雑性、言い換えると表現力を決定する重要なパラメータである。他のハイパーパラメータも含めてハイパーパラメータ探索を実行してしまうと、限られた試行回数では最適な N_x を見つける前に探索が終わってしまう可能性が高い。また、 N_x は式(2.4)や(2.40)からわかる通り、他のハイパーパラメータの決定にも影響を与える。以上から、本研究では N_x だけ個別に最適化する。具体的には、 N_x を区間[5, 200]の間で変化させていき、 N_x を固定した後で他のハイパーパラメータを 512 回のベイズ最適化で探索する。ここで、計算コスト削減のために、区間[5, 200]は[5, 50]までは 1 刻み、[50, 200]は 10 刻みで変化させている。 N_x に対する予測誤差 NRMSE を求めた。ここで、NRMSE はテスト区間のタイムステップ（図 4.1, 図 4.2 の Test と表記した区間）の値とし、この区間のデータをテストデータと定義する。ただし、放電電圧 V_d または推進剤流量 $\dot{m}$ が変化しているサイクルは除いて算出している。以降、テストデータと表記している場合はこの区間のことを指す。ここでは、目標出力が放電電流 I_d のときの結果を示している。目標出力が V_{cg} の結果は付録 F で示す。以下に、データセットごとの結果を示す。

[データセット A]

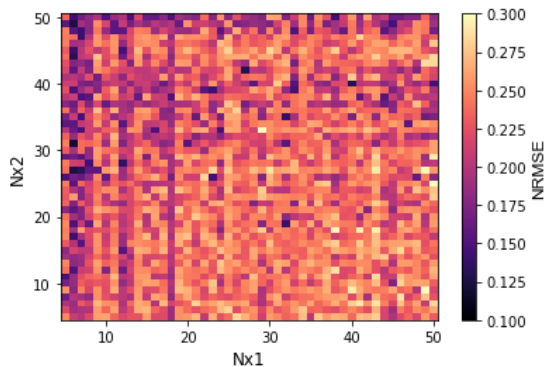
N_x に対する予測誤差 NRMSE のマップを図 5-2 に示す. simpleESN は, $N_x = 9$ のときに NRMSE が最小値をとる. $N_x = 90$ 以降で急激に NRMSE が大きくなっているのが確認できる. これはモデルの表現力が高くなり過ぎて, 過学習が発生しているからだと考えられる.

Cycle-groupedESN は $(N_{x1}, N_{x2}) = (50, 70)$ のときに最小値をとり, 区間 $[50, 200]$ では $(70, 150)$ である. 図 5-2(b) より $[5, 50]$ の区間では, N_{x1} が 5~8 の領域で NRMSE が 0.15 以下と (N_{x1}, N_{x2}) の組が多く存在する. 図 5-2(c) より $[50, 200]$ の区間では, NRMSE が 0.15 以下となる領域がマップ全体に広く分布している.

(a)



(b)



(c)

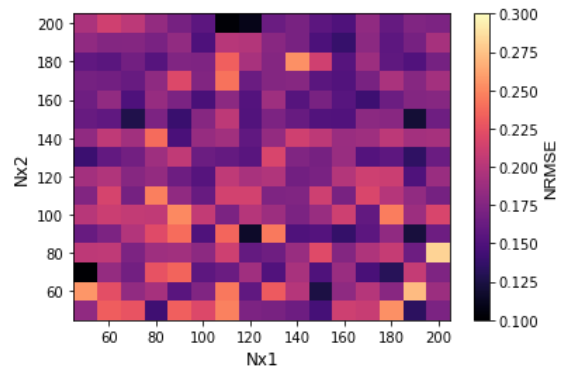
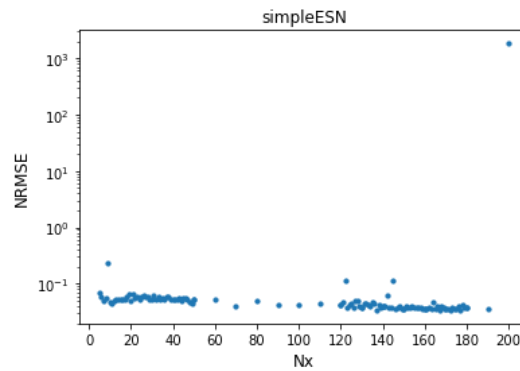


図 5-2 N_x に対する NRMSE マップ (データセット A) (a) simpleESN
(b) cycle-groupedESN 区間 $[5, 50]$ (c) cycle-groupedESN 区間 $[50, 200]$

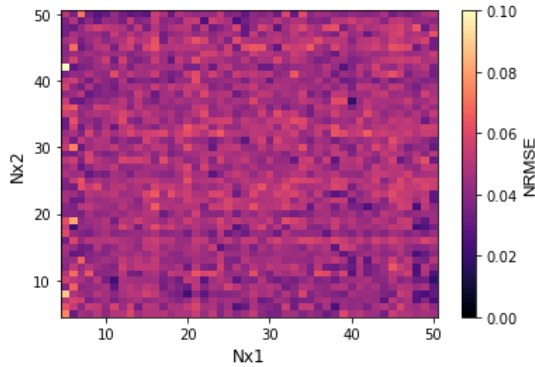
[データセット B]

N_x に対する予測誤差 NRMSE のマップを図 5-3 に示す. 図 5-3(a)に示すように simpleESN は $N_x = 200$ で急激な NRMSE の増加が起きている. 区間[120, 180]付近で再度計算を行い図 5-3(b)を得て, $N_x = 167$ のときに NRMSE が最小値をとることがわかった. 図 5-3(c)(d)より, Cycle-groupedESN は $(N_{x1}, N_{x2}) = (30, 28)$ のときに最小値をとり, 区間[50, 200]では(90, 130)であった. しかしながら, マップ上のほとんどの領域で NRMSE が 0.1 以下となるため, 任意に (N_{x1}, N_{x2}) を決めたとしても NRMSE が小さくなると予想される. これはモデル設計時に最適な (N_{x1}, N_{x2}) を探す手間を省くことにつながる.

(a)



(b)



(c)

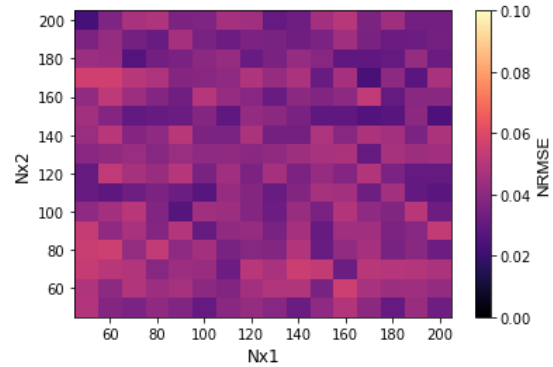


図 5-3 N_x に対する NRMSE マップ (データセット B) (a) simpleESN
(b) cycle-groupedESN 区間[5,50] (c) cycle-groupedESN 区間[50, 200]

[まとめ]

データセットが異なれば最適な N_x のマップの特徴もが変化するのが確認できた.
各データセットの共通点は

1. simpleESN はデータセットによって最適な N_x にばらつきがある. 一定以上の大きさ(データセット A は $N_x = 90$, データセット B は $N_x = 200$)になると, 急激に NRMSE が大きくなる.
2. cycle-groupedESN の場合, NRMSE が小さい(N_{x1}, N_{x2})が区間[50,200]のマップ上の広い領域に存在する. したがって, 区間[50,200]であれば, 任意に N_x を決めても NRMSE が小さい可能性が高い.

5-3. ベイズ最適化の結果

リザーバーのノード数 N_x 以外のハイパーパラメータは3-2節のベイズ最適化を512試行することで最適化されたハイパーパラメータを決定している。本節では、データセット A を用いて目標出力の放電電流 I_d を予測する場合に絞って、ベイズ最適化の結果を議論する。実際に、PPU の計算機でベイズ最適化を行う場合は1回のハイパーパラメータ探索で十分である。しかし、それでは ESN モデルを用いた推進性能予測タスクにおけるベイズ最適化の特性を得ることが難しい。そこで、1探索512試行のベイズ最適化を512探索行うことで、ベイズ最適化の結果をサンプル数512の分布として得ることにした。ここで、乱数シードを各探索で異なるものに設定している。乱数シードとは疑似乱数を初期化するときの数値である。乱数シードを固定すれば生成される乱数も固定され、乱数シードを変えると生成される乱数も変わる。以降は本予測タスクのベイズ最適化の履歴、ハイパーパラメータの重要度 importance, cycle-groupedESN のハイパーパラメータの分布を見ていく。

まず、ベイズ最適化の探索履歴を図 5-4 に示す。simpleESN($N_x = 9$)と cycle-groupedESN($(N_{x1}, N_{x2}) = (6, 31)$)の2つ比較した。図 5-4 の横軸は試行回数(512回)、縦軸がベイズ最適化区間の NRMSE である。この NRMSE はテストデータの NRMSE とは異なり、ベイズ最適化中に算出される1~16400ステップ間の NRMSE である。各試行で、512探索分の分布が得られるので、ノンパラメトリックな方法で中央値と両側95%予測区間を算出した。図 5-4 からわかる通り、どちらのモデルも50試行あたりの早い段階で、NRMSE が収束している。また、simpleESN の方が両側95%予測区間が狭く、初期条件に依らず最終的には NRMSE = 0.0651 に近い値に収束することがわかる。

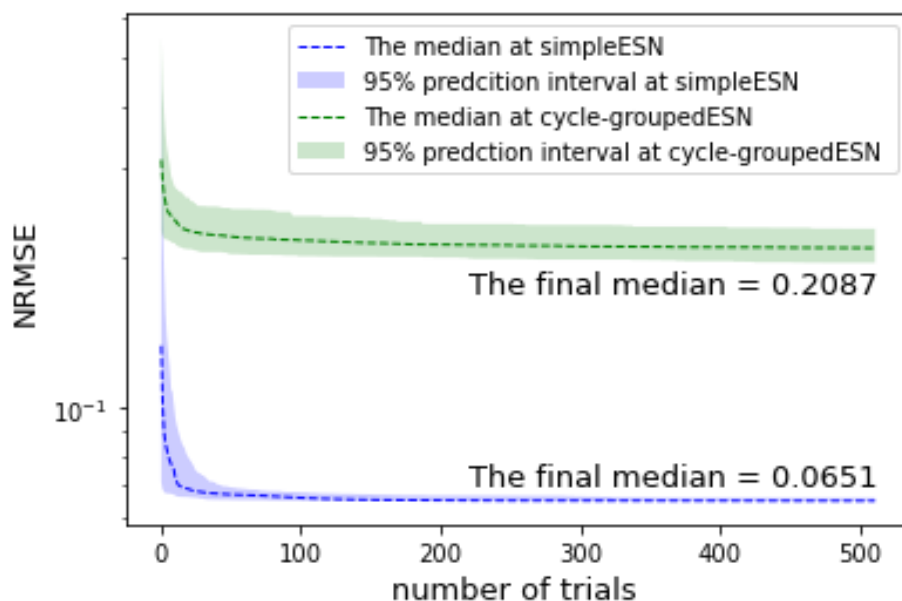


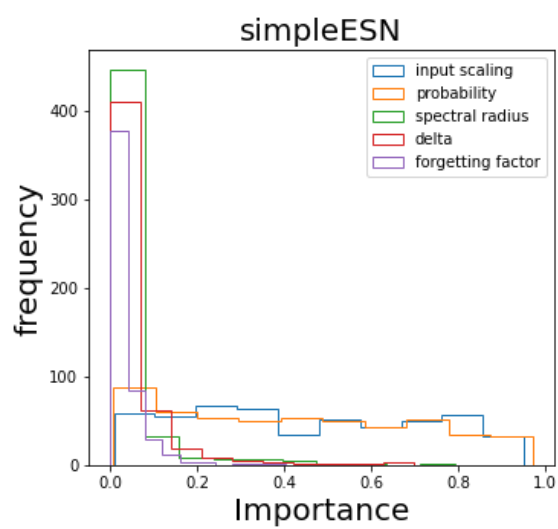
図 5-4 ベイズ最適化の探索履歴

次に、各ハイパーパラメータの調節が予測精度の向上に寄与したかを調査した。これはハイパーパラメータの **importance** という指標で評価される。対象のハイパーパラメータの **importance** はすべてハイパーパラメータの **importance** に対しての割合で表現されるため、0 から 1 の間の数値をとる。**importance** が大きいハイパーパラメータほど、そのパラメータの調節次第で予測精度が大幅に変わることを意味する。したがって、**importance** が上位のハイパーパラメータは、適切な値の設定がモデル設計時に必要になる。fANOVA というアルゴリズム[47]を用いて、本実験の **importance** は導出している。図 5-5 に 512 探索して得たハイパーパラメータの **importance** の分布を示す。図 5-5(a)が simpleESN, 図 5-5(b)が cycle-groupedESN の結果である。simpleESN は入力スケーリング a_{in} と結合確率 p が **importance** > 0.5 以上となる回数が多い。また、cycle-groupedESN はリザバー 1 の入力スケーリング a_{in1} が **importance** > 0.5 となる回数が多い。したがって、今回のベイズ最適化では simpleESN は a_{in} と p が **importance** 上位のハイパーパラメータ、cycle-groupedESN は a_{in1} が **importance** 上位のハイパーパラメータであるといえる。

a_{in} が式(2.4)の入力結合重み W^{in} のスケーリングを決定するため、新たなリザバー状態ベクトル $\mathbf{x}(n)$ に対する既存のリザバー状態ベクトル $\mathbf{x}(n-1)$ と新たな入力 $\mathbf{u}(n)$ の影響度合いを決定する。結合確率 p はリザバーのスパース性を決定し、モデルの表現力の多様性を決定するため、 p も **importance** 上位のパラメータになったと考える。

ただし、この **importance** の解釈には注意が必要である。本研究の各ハイパーパラメータの探索空間は任意に決定した。この場合、モデル性能が大幅に向上する領域と悪化する領域の 2 つが共存した探索空間をもつハイパーパラメータと、モデル性能がほとんど探索空間内ではほとんど変化しないハイパーパラメータの 2 通りが存在する可能性が高い。前者は **importance** が大きくなる一方で、後者は小さくなる。しかし、後者の探索空間を調節することでモデル性能が大幅に向上(もしくは悪化)する領域も含むようになり、**importance** も大きくなるのではないかと考える。したがって、探索空間を変える、調節するハイパーパラメータを現状の 5 つから減らす、といった方法を用いた追加の調査が必要である。

(a)



(b)

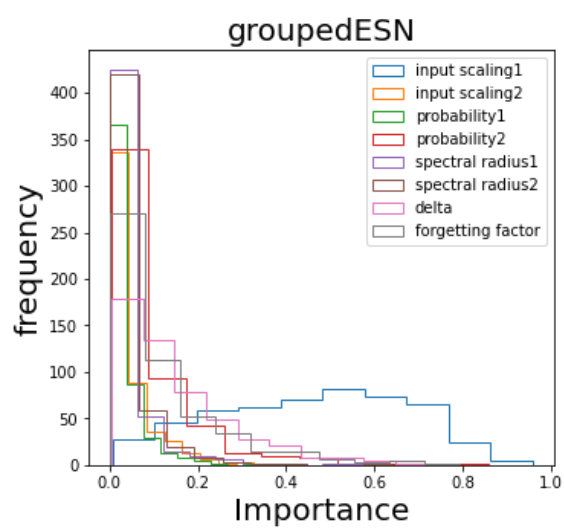


図 5-5 importance の分布 (a) simpleESN (b) cycle-groupedESN

次に、cycle-groupedESN の最適化されたハイパーパラメータに、リザバー 1 とリザバー 2 で違いが見られるか確認した. 図 5-6 に入力スケール a_{in} , 結合確率 p , スペクトル半径 $\rho(W)$ の 3 つのハイパーパラメータの分布を示す.

図 5-6(a)より, 入力スケール a_{in1} は探索空間 $[0.0001, 0.01]$ の上限, a_{in2} は探索空間の下限周辺に分布が偏っている. どちらも探索空間の範囲外に分布の裾が広がっている可能性が高い. この結果から, 式(5.1)(5.2)で示したリザバー 1 の入力 $u_1(m)$ はリザバー 2 の入力 $u_2(m)$ よりも活性化関数 $\tanh(-)$ の非線形の強い領域での変換が適していることがわかる. a_{in} と非線形性の関係は付録 E を参照されたい.

図 5-6(b)より結合確率 p もリザバー 1 とリザバー 2 で分布に違いは見られる. p_2 は p_1 に比べて密なりザバー状態を好む. 一方, 図 5-6 からスペクトル半径 $\rho(W)$ は, $\rho_1(W)$ と $\rho_2(W)$ の両者とも探索空間内に一様に分布している傾向が見られる. この傾向が探索空間を広げても同様なのかはさらなる検証が必要である.

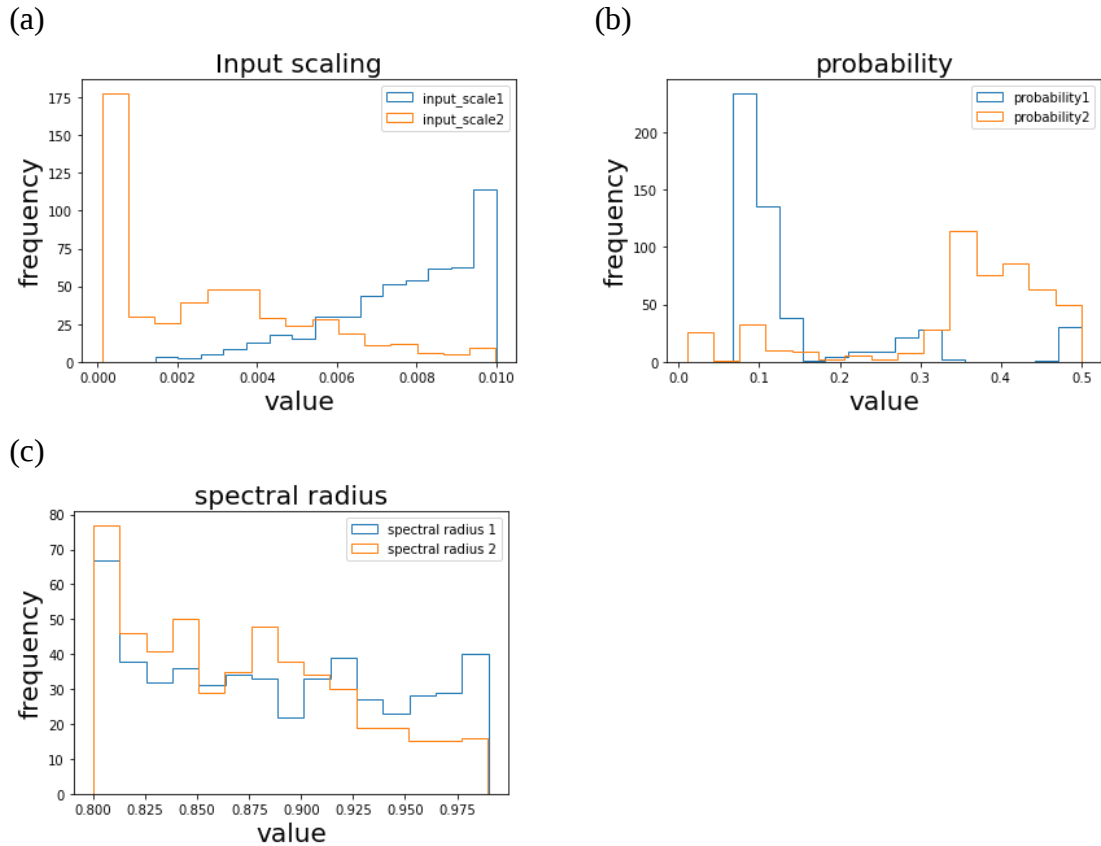


図 5-6 cycle-groupedESN の最適ハイパーパラメータ分布 (a) a_{in} (b) p (c) $\rho(W)$

5-4. 探索するデータ区間

5-2 節と 5-3 節のベイズ最適化は、図 4-1~4-2 で Bayesian optimization と示した区間のデータを用いている。データセット A は 16400 ステップまでのデータである。ESN モデルの空間計算量は $O(N_x^2 T)$ であり、 T は学習につかうタイムステップ数である[34]。したがって、学習させるタイムステップ数を減らすことで計算時間の短縮に繋がる。そこで、本節ではデータセット A を用いて学習させるデータ区間を減らした場合に予測精度 NRMSE がどういう反応を示すかを調査した。データセット A では、放電電圧が切り替わるタイミングである 3600, 6400, 10000, 13600, 16400 ステップ目でデータを区切り 1 ステップ目から各区切りステップまでをひとつのデータセットとした。つまりタイムステップ数が 3600, 6400, 10000, 13600, 16400 の 5 パターンのデータセットを用意した。モデルは simpleESN と cycle-groupedESN の 2 種類を使用し、乱数シードを変えながら 512 探索のベイズ最適化を 10 探索ずつ行った。5-2 節の調査での NRMSE が最小値だったノード数 N_x を採用し、simpleESN は $N_x = 9$, cycle-groupedESN は $(N_{x1}, N_{x2}) = (50, 70)$ に設定している。

表 5-3 に simpleESN モデルの 10 点分のテストデータの NRMSE の中央値と標準偏差、NRMSE の平均を示す。NRMSE の中央値に差は見られないが、標準偏差がタイムステップ数によって異なる。これはタイムステップ数が 3600 のときに $\text{NRMSE} = 3.24$, 13600 のときに $\text{NRMSE} = 0.588$ の外れ値を含むためである。外れ値の場合も探索自体は上手くいったため、この外れ値は予測時に発生するものだと考える。ホールスラストの電源制御においては、予期せぬ外れ値による誤作動は避けなければならない。

次に、表 5-4 に cycle-groupedESN モデルの結果を示す。こちらはタイムステップ数が増えるにつれて NRMSE が小さくなっている。これは、RLS アルゴリズムによる出力結合重み W^{out} の更新頻度に起因していると考ええる。simpleESN は 1 サイクルで 400 回更新するのに対して、cycle-groupedESN は 1 サイクルで 1 回の更新である。したがって、cycle-groupedESN はデータ数が増えることで更なる NRMSE の減少が見込める。標準偏差に関しては、simpleESN のタイムステップ数(3600, 13600)で見られた外れ値が cycle-groupedESN では存在しないため、どのタイムステップ数も 10 の -2 乗オーダーの値をとっている。しかしながら、この値は simpleESN の外れ値のないタイムステップ数(6400, 10000, 16400)の場合と比べて一桁大きい。

また、図 5-7 にタイムステップ数と NRMSE の関係を示した図を示す。横軸がタイムステップ数であり、縦軸が NRMSE である。ただし、simpleESN のタイムステップ数が 3600 のときの $\text{NRMSE} = 3.24$, 13600 のときの $\text{NRMSE} = 0.588$ の外れ値は表示していない。cycle-groupedESN の方が simpleESN よりもばらついてるのがわかる。ただし、simpleESN は高い頻度(10 回中 0~1 回)で、予測が上手くいかないケースがあるため、安定性の観点では cycle-groupedESN が優れているといえる。

表 5-3 ベイズ最適化に用いるタイムステップ数を変えたときの
NRMSE と探索時間
(データセット A, simpleESN, $N_x = 9$)

タイムステップ数	NRMSE の中央値	NRMSE の標準偏差	探索時間 [秒]
3600	0.0459	9.59×10^{-1}	83
6400	0.0413	1.33×10^{-3}	139
10000	0.0422	1.77×10^{-3}	210
13600	0.0427	1.63×10^{-1}	283
16400	0.0407	5.66×10^{-3}	339

表 5-4 ベイズ最適化に用いるタイムステップ数を変えたときの
NRMSE と探索時間

(データセット A, cycle-groupedESN, $(N_{x1}, N_{x2}) = (50, 70)$)

タイムステップ数	NRMSE の中央値	NRMSE の標準偏差	探索時間 [秒]
3600	0.2624	2.63×10^{-2}	20
6400	0.2512	3.60×10^{-2}	22
10000	0.1949	1.82×10^{-2}	24
13600	0.1871	2.04×10^{-2}	27
16400	0.1773	3.20×10^{-2}	29

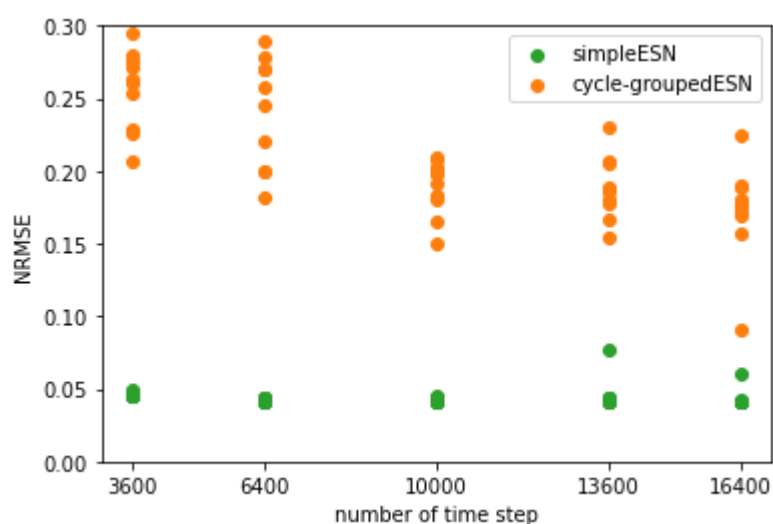


図 5-7 ベイズ最適化に用いるタイムステップ数と NRMSE の関係

5-5. Long Short-Term Memory (LSTM)モデル

ESN モデルの比較として, LSTM モデルを用意した. この LSTM モデルも, 1 つデータが加わるたびに学習パラメータを更新するオンライン学習を行った. LSTM の入力 $\mathbf{u}(n)$ は,

$$\mathbf{u}(n) = [V_d(n-9), \dots, V_d(n), \dot{m}(n-9), \dots, \dot{m}(n), I_{\text{coil}}(n-9), \dots, I_{\text{coil}}(n)]^T \in \mathbb{R}^{30} \quad (5.6)$$

と表現しており, これは現在のタイムステップ n から遡って 10 ステップの作動条件を入力していることを意味する. この遡るタイムステップ数(今回は 10)は window size と呼ばれ, LSTM のハイパーパラメータである. 最適な window size の調節も必要だが, LSTM のハイパーパラメータは 1 試行で膨大な計算時間がかかるため本研究では行っていない. 事前に window size が 10 と 100 の 2 パターンを試し, NRMSE が小さかった window size = 10 を採用している. 表 5-5 に LSTM の本実験の設定を示す. 隠れ層のユニット数を増やせば, リザーバーのノード数 N_x と同様表現力は高まるが計算コストも増えてしまう. ドロップアウト率は, 過学習を抑制するためにユニット間の信号を遮断する確率のことを指す. 学習アルゴリズムは Adam (Adaptive moment)[48]を採用した.

表 5-5 LSTM の諸設定

ハイパーパラメータ	
window size	10
隠れ層のユニット数	50
ドロップアウト率	0.1
学習アルゴリズム	Adam

5-6. 放電電圧 V_d が変化するときの予測結果

データセット A を用いたときの予測結果を示す．データセット A は，推進剤流量を 7sccm で固定し，放電電圧を変化させている．詳しくは図 4-1 を参照する．

予測結果は simpleESN，cycle-groupedESN，LSTM の 3 種類のモデルのモデル出力を比較している．目標出力 $\mathbf{d}(n)$ は推進性能の放電電流 I_d と接地電位に対するカソード電位 V_{cg} の 2 種類である．入力 $\mathbf{u}(n)$ は作動条件（推進剤流量 $\dot{m}$ ，放電電圧 V_d ，コイル電流 I_c ）とした．ここで，推進性能と作動条件のすべては平均 0，分散 1，となる標準化をデータの预处理で施している．

以下で，目標出力が放電電流 I_d かカソード電圧 V_{cg} の場合に分けて予測結果を提示する．

[目標出力：放電電流 I_d]

ノード数は 5-2 節の N_x と NRMSE のマッピング結果から，NRMSE が最小だった N_x を採用している．simpleESN は，[5, 50]の範囲から $N_x = 9$ ，[50, 200]の範囲から $N_x = 60$ を採用し予測結果を比較した．同様に，cycle-groupedESN のノード数は，[5, 50]の範囲から $(N_{x1}, N_{x2}) = (6, 31)$ ，[50, 200]の範囲から $(N_{x1}, N_{x2}) = (50, 70)$ を採用している．他のハイパーパラメータは 512 試行のベイズ最適化 1 探索で最適化した．

表 5-6 に各モデルのテストデータに対する NRMSE と学習時間を示している．ESN の学習時間は，1~16400 ステップのデータを用いて 512 試行のベイズ最適化した計算時間を含んでいる．LSTM はすべてのタイムステップを 1 度きりオンライン学習を行った時間である．simpleESN と cycle-groupedESN は両者とも LSTM よりも NRMSE が小さく， $N_x = 9$ のときの simpleESN の NRMSE が一番小さい．学習時間に関しては，cycle-groupedESN はベイズ最適化なしの LSTM よりも 2 桁，simpleESN よりも 1 桁のオーダーで小さい．これは cycle-groupedESN の学習に用いるタイムステップ数 T が simpleESN の 1/400 であるからである．simpleESN の空間計算量は $\mathcal{O}(N_x^2 T)$ ，cycle-groupedESN は $\mathcal{O}((N_{x1} + N_{x2})^2 T)$ と表されるため[34]，ノード数 N_x と総タイムステップ数 T が計算時間を決定する要因となる．本研究のデータセットはサンプリング周波数 40Hz だった．より高いサンプリング周波数のデータ，simpleESN と cycle-groupedESN の計算時間の差はより顕著になると考えられる．

図 5-8(a) に 19600~21200 ステップ，図 5-8 (b)に 20800~21200 ステップでの，目標出力(黒色)と simpleESN(緑色とオレンジ色)と LSTM(青)のモデル出力をプロットする．simpleESN は LSTM と比較して当てはまりがいいのがわかる．また，図中に赤丸で示した放電電流の急激な揺らぎを予測することが出来ている．しかし，ノード数 $N_x = 60$ の場合は， $N_x = 9$ のときと比較して，モデル出力が細かく振動しているのがわかる．これは，データの複雑性に対して，モデルの複雑性が大きくなり過ぎる過学習現象が起きているからだと考えられる．

同様に，図 5-8 (b) (c) に cycle-groupedESN のモデル出力をプロットする．19600~20000 ステップの間は放電電圧が変化しているサイクルなので，当てはまりが悪い．放電電圧の遷移が終わった 20000 ステップ以降のサイクルは，徐々に当てはまりが良くなっている．これは RLS アルゴリズムによるオンライン学習の特徴である．データが入ってくるたびに学習パラメータを更新するため，徐々に予測精度が向上していく．

表 5-6 目標出力 I_d のときの予測結果の比較(データセット A)

モデル	N_x (N_{x1}, N_{x2})	NRMSE	計算時間[秒]
simpleESN	9	0.0407	303
simpleESN	60	0.0584	638
cycle-groupedESN	6, 31	0.1179	24
cycle-groupedESN	50, 70	0.0912	39
LSTM	-	0.4640	2533

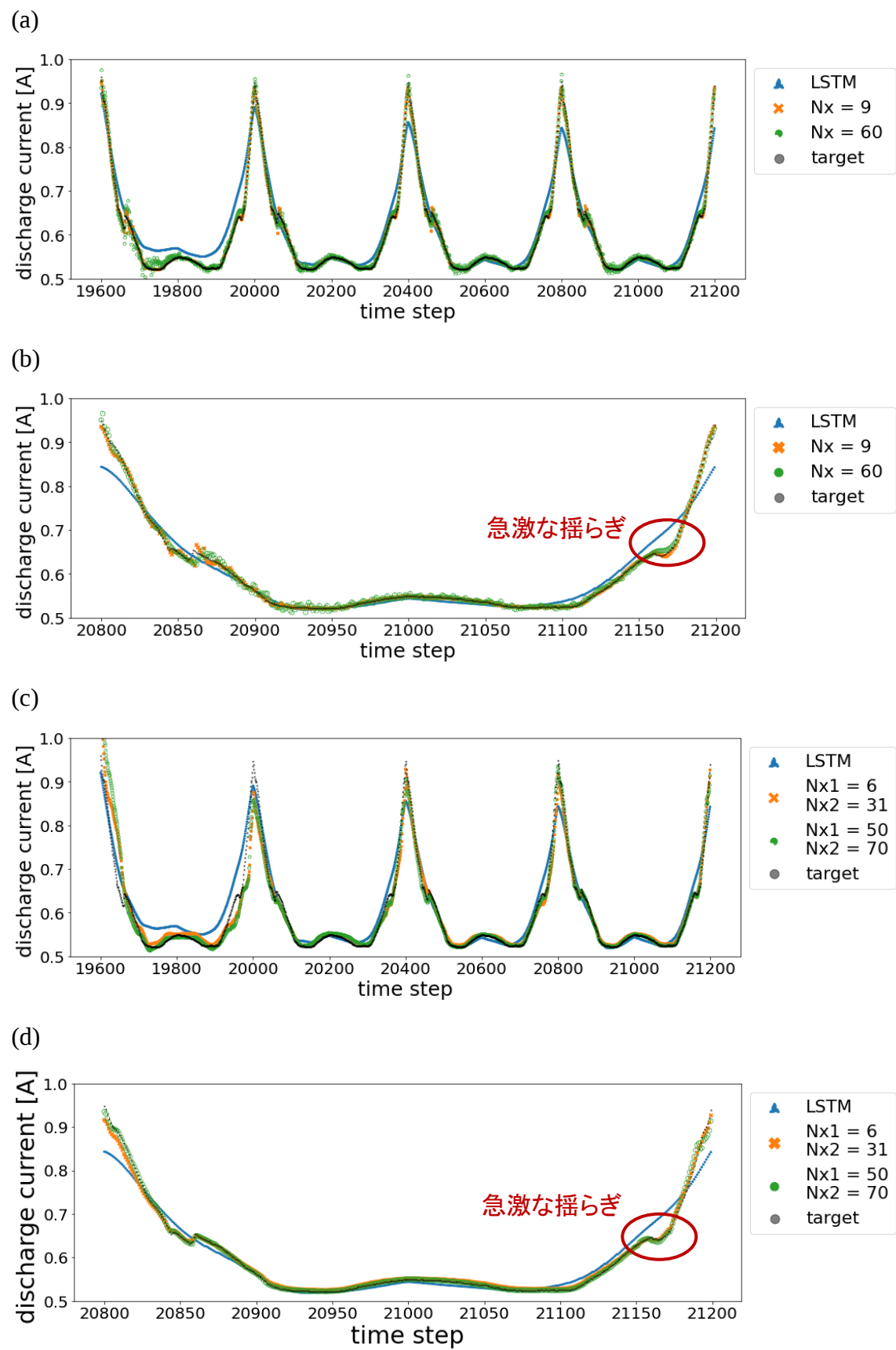


図 5-8 目標出力 I_d とモデル出力の比較(データセット A)
 (a) (b) simpleESN (c) (d) cycle-groupedESN

[目標出力：カソード電位 V_{cg}]

simpleESN は、[5, 50]の範囲から $N_x = 11$ ，[50, 200]の範囲から $N_x = 60$ を採用した．同様に，cycle-groupedESN は、[5,50]の範囲から $(N_{x1}, N_{x2}) = (7, 45)$ ，[50, 200]の範囲から $(N_{x1}, N_{x2}) = (50, 70)$ を採用した．

表 5-7 に各モデルのテストデータに対する NRMSE と学習時間を示している．目標出力が放電電流 I_d のときと同様，どちらの ESN モデルの NRMSE は LSTM よりも小さく， $N_x = 9$ の simpleESN が一番小さいモデルである．計算時間に関しては，cycle-groupedESN が一番小さいという結果になった．

図 5-9 (a) に simpleESN モデル出力をプロットする．放電電流 I_d のときと同様に，simpleESN は LSTM と比較して当てはまりがよいのがわかる．ノード数 $N_x = 11$ のときに一番当てはまりが良い．ただ， $N_x = 11$ の場合でも，図中の赤丸で示す 20100, 20500, 20900 ステップ付近の V_{cg} が立ち上がる場所の予測の当てはまりが悪い．

同様に，図 5-9 (b) に cycle-groupedESN のモデル出力をプロットする．こちらも放電電流を予測した結果のときと同様にサイクルが増えるにつれて，当てはまりがよくなっている．ただし，21070 ステップ付近の V_{cg} が急激に下がり一定となる区間で，両方の ESN モデル出力がアンダーシュートする現象が見られた．

表 5-7 目標出力 V_{cg} のときの予測結果の比較(データセット A)

モデル	N_x (N_{x1}, N_{x2})	NRMSE	計算時間[秒]
simpleESN	11	0.0981	305
simpleESN	60	0.3227	598
cycle-groupedESN	7, 45	0.3690	26
cycle-groupedESN	50, 70	0.4225	37
LSTM	-	0.6832	2756

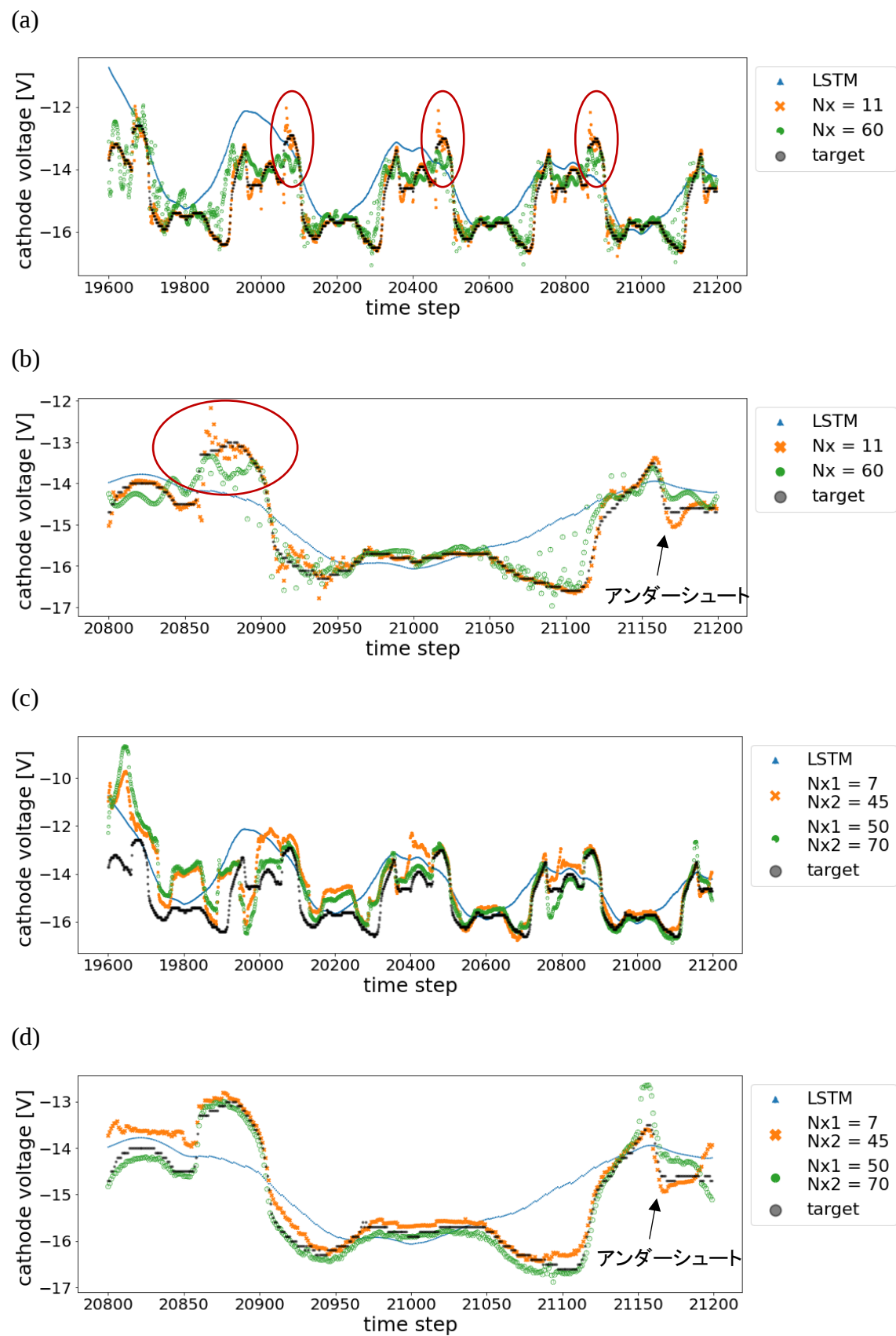


図 5-9 目標出力 V_{cg} とモデル出力の比較(データセット A)
 (a) (b) simpleESN (c) (d) cycle-groupedESN

5-7. 推進剤流量 m が変化するときの予測結果

データセット B を用いたときの予測結果を示す．データセット A は，放電電圧を 300 V で固定し，推進剤流量を変化させている．詳しくは図 4-2 を参照．データセット以外は 5-6 節と同一の条件での結果を以下で示すため，条件設定等の説明は最小限に留める．

[目標出力：放電電流 I_d]

表 5-8 にテストデータの NRMSE と学習時間を示す．テストデータは図 4-2 のから 11200~24000 ステップである．学習時間は，1~10400 ステップのデータを用いて 512 試行のベイズ最適化を行った計算時間を含んでいる．ESN モデルは両者とも LSTM より NRMSE が小さく， $(N_{x1}, N_{x2}) = (50, 10)$ の cycle-groupedESN の NRMSE が一番小さい．また，計算時間もこの条件のモデルが一番小さく，512 試行の探索を 21 秒で終わることが出来ている．

図 5-10(a)に simpleESN，図 5-10(b)に cycle-groupedESN のモデル出力のプロットを示す．目標出力 I_d はパルスの立ち上がり前と立ち下がり後に放電電流が沈み込む現象が見られる(図 5-10 の赤印)．simpleESN と cycle-groupedESN は両者ともこの沈み込みを予測することが出来ている．cycle-groupedESN が simpleESN よりも精度良く予測できたのは，各サイクルでの沈み込みの形が似ていたからだと考える．

表 5-8 目標出力 I_d のときの予測結果の比較(データセット B)

モデル	N_x (N_{x1}, N_{x2})	NRMSE	計算時間[秒]
simpleESN	49	0.0445	241
simpleESN	167	0.0340	1132
cycle-groupedESN	50, 10	0.0176	21
cycle-groupedESN	50, 200	0.0224	66
LSTM	-	0.2803	1695

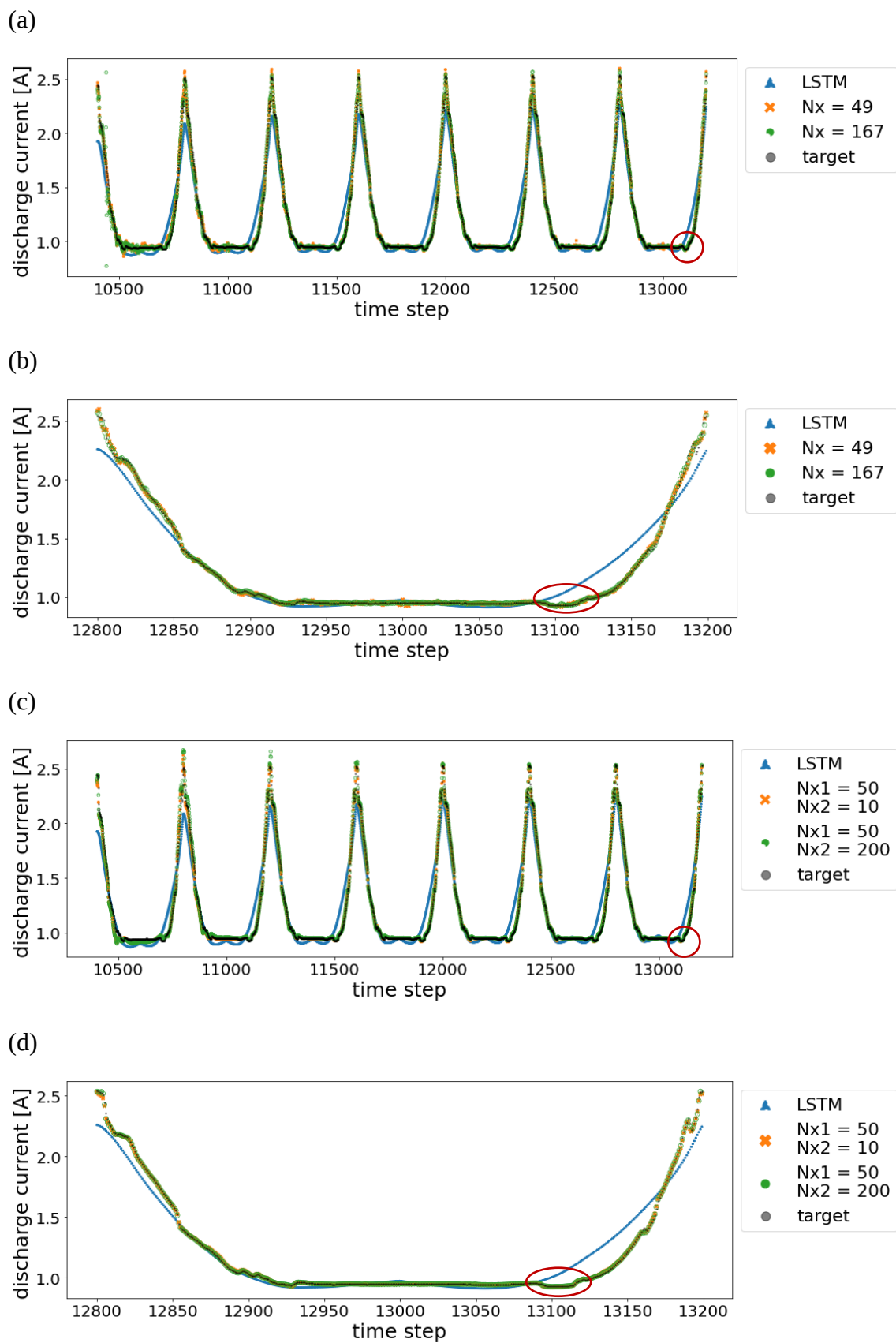


図 5-10 目標出力 I_d とモデル出力の比較(データセット B)

(a) (b) simpleESN (c) (d) cycle-groupedESN

[目標出力：カソード電位 V_{cg}]

表 5-9 に各モデルのテストデータに対する NRMSE と学習時間を示している. cycle-groupedESN を用いて $(N_{x1}, N_{x2}) = (50, 150)$ に設定した時が, NRMSE が一番小さい.

図 5-11 に simpleESN と cycle-groupedESN のモデル出力と目標出力 V_{cg} をプロットする. 10400~11200 ステップまでは, 推進剤流量が増加しているサイクルである. cycle-groupedESN はこのサイクル間の予測は上手くいっていないが, 推進剤流量が一定となるとサイクルを重ねるごとに当てはまりが良くなった. また, ESN のどちらのモデルも, V_{cg} が -50 V 付近まで大きく沈みこむ現象も予測することが出来た. カソードの損耗を抑制するためには, このような沈み込んだ領域での作動は避けなければならない. したがって, この沈み込みを作動条件のみで表現できる予測モデルの構築は, モデルの良し悪しを決める重要な指針である.

表 5-9 目標出力 V_{cg} のときの予測結果の比較(データセット B)

モデル	N_x (N_{x1}, N_{x2})	NRMSE	計算時間[秒]
simpleESN	44	0.2077	244
simpleESN	174	0.1392	1249
cycle-groupedESN	6, 44	0.0550	20
cycle-groupedESN	50, 150	0.0533	50
LSTM	-	0.9261	2071

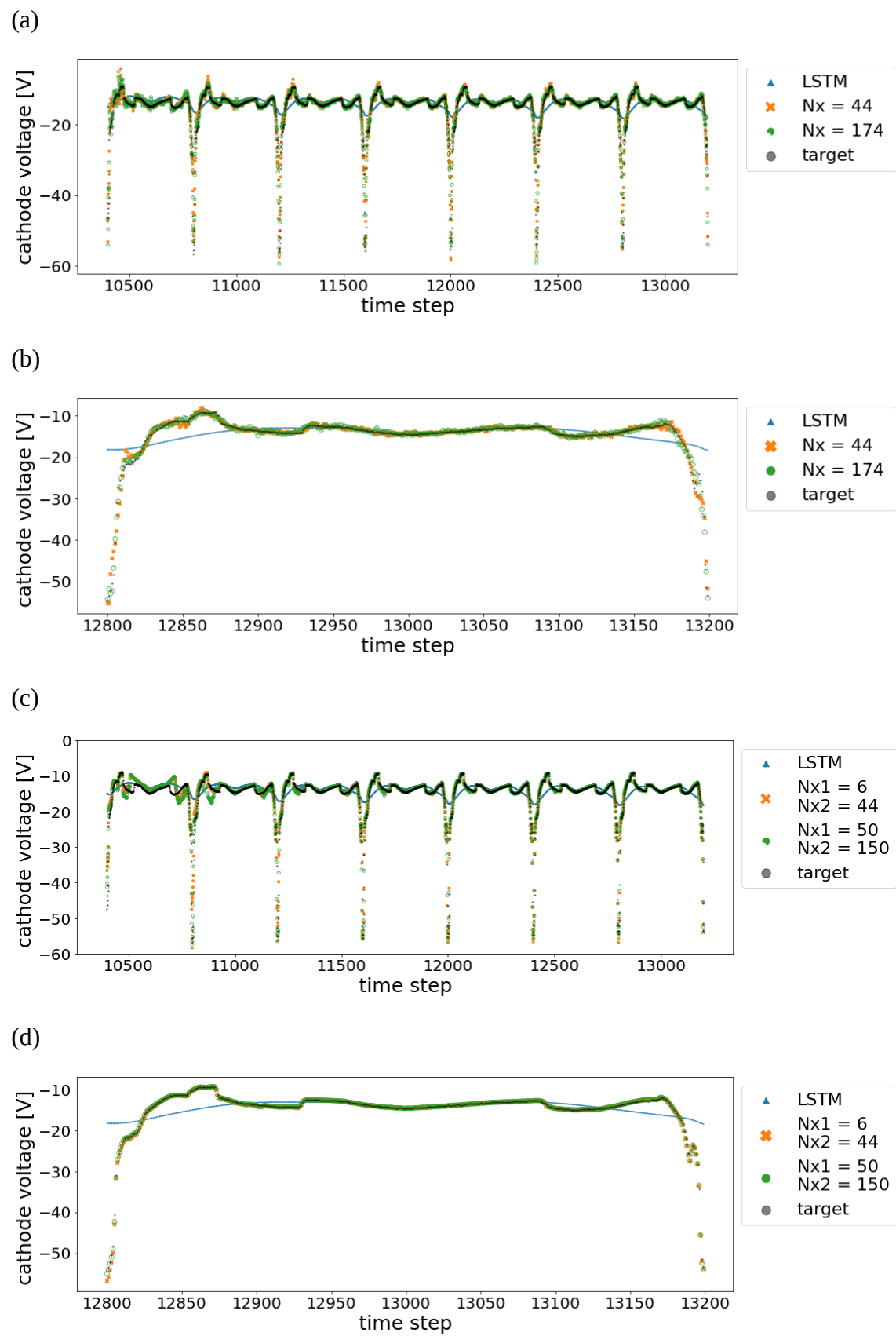


図 5-11 目標出力 V_{cg} とモデル出力の比較(データセット B)

(a) (b) simpleESN (c) (d) cycle-groupedESN

5-8. simpleESN と cycle-groupedESN の比較考察

本節では、5-2~5.8 節で得られた結果をもとに、実際に SA-PPU の最適作動条件下での電源制御に simpleESN と cycle-groupedESN を用いた際のそれぞれのメリット、デメリットを考察する。

[simpleESN]

5-6 節の表 5-6, 表 5-7 からわかる通り、高精度の推進性能予測が可能であることが利点である。ただし、5-2 節の図 5-2 で示すように、 N_x の最適化が上手くいかないと急激に予測精度が悪くなってしまう。

また、図 5-2 と図 5-3 の比較より、用いるデータセットによって最適な N_x が大きく異なることがわかる。データセット A の場合は $N_x = 9$ 、データセット B の場合は $N_x = 167$ である。この結果は、新しいデータセットを用意した場合には、その都度 N_x の最適化が必要になることを示唆している。

まとめると、simpleESN は高精度の予測を可能にするモデルだが、ハイパーパラメータの設定を厳密に行う必要があるモデルでもある。

[cycle-groupedESN]

データセット B のように、各サイクルの波形が似た形の場合は、simpleESN より予測精度が良いと結果を得た(5-7 節の表 5-8, 表 5-9)。また、simpleESN に比べて計算時間が 1/10 程度というのも大きな利点である。この計算時間の差は、1 サイクル内のデータ点数が増えればより顕著になる。

5-2 節の図 5-2 と図 5-3 より、cycle-groupedESN は (N_{x1}, N_{x2}) が比較的大きくても simpleESN で見られた予測精度の悪化みられないことがわかる。また、NRMSE 小さい領域がマップ上に広く分布しているため、厳密な (N_{x1}, N_{x2}) の設定をしなくても、ある程度の予測精度を保てることが考えられる。

まとめると、cycle-groupedESN は計算時間の観点で優れており、用いるデータセットによっては simpleESN よりも高精度の予測が可能である。また、ハイパーパラメータの設定において、simpleESN ほどの厳密さは求められないと考えられる。

第6章 結論

本研究では、200W 級ホールスラストの推進性能予測において、再帰型ニューラルネットワーク(RNN)の Echo State Network (ESN) モデルが有効であるかを評価した。評価指標は予測精度と計算コストの 2 点である。比較対象は同じく RNN の Long Short-Term Memory(LSTM)とし、ESN モデルは通常の simpleESN とリザバーを 2 層並列に並べ 1 サイクル分の情報を一度に入力する cycle-groupedESN の 2 種類を用意した。また、ESN モデルの設計は、リザバーのノード数 N_x の最適化マップの作成およびベイズ最適化を用いたハイパーパラメータ探索を組み合わせることで、予測精度の向上を図った。予測に用いるデータセット A は放電電圧、データセット B は推進剤流量が一定時間ごとに変化するデータであり、推進性能の放電電流 I_d もしくは接地電位に対するカソード電位 V_{cg} を目標出力として予測した。以下に、得られた結果をまとめる。

1. リザバーのノード数 N_x の最適化マップを作成した。simpleESN はデータセット A の場合、 I_d 予測で $N_x = 9$ 、 V_{cg} 予測で $N_x = 11$ のときに誤差を示す正規化平方平均二乗誤差(NRMSE)が最小となった。一方でデータセット B の場合は、 I_d 予測で $N_x = 167$ 、 V_{cg} 予測で $N_x = 174$ となった。これはデータセットに依って最適な N_x が大きく変化してしまうことを示唆する。Cycle-groupedESN の NRMSE が小さくなる(N_{x1}, N_{x2})の組み合わせは、区間[50, 200]のマップ全体に広く分布しているのが確認できた。これはデータセット A, B に共通した特徴である。(5-2 節)
2. 512 試行のベイズ最適化を 512 探索行った結果、入力スケーリング a_{in} とリザバーのリカレント結合重み W の非ゼロ要素確率 p の 2 パラメータが、NRMSE の減少に大きく寄与しているパラメータだと判明した。この結果は、他のデータセット等で更なる検討が必要だと考える。(5-3 節)
3. ベイズ最適化に用いるタイムステップ数を変化させたときの NRMSE を取得した。simpleESN はタイムステップ数を変化させても NRMSE に変化は見られなかった。一方で cycle-groupedESN はタイムステップ数を増やすことで NRMSE の減少が確認できた。(5-4 節)
4. テストデータに対するモデルの予測結果から、simpleESN, cycle-groupedESN の両者は LSTM に比べて NRMSE が小さいという結果を得た。また、512 試行のベイズ最適化を行った ESN モデルの計算時間はハイパーパラメータ探索を行っていない LSTM より小さかった。(5-6.7 節)

5. simpleESN と cycle-groupedESN を比較した。データセット A の場合は simpleESN の方が予測精度は良かった。 I_d 予測のとき $\text{NRMSE} = 0.0407$, V_{cg} 予測のとき $\text{NRMSE} = 0.0981$ だった。一方で、データセット B の場合は cycle-groupedESN の方が予測精度は良かった。 I_d 予測のとき $\text{NRMSE} = 0.0176$, V_{cg} 予測のとき $\text{NRMSE} = 0.0533$ だった。計算時間を比較すると、cycle-groupedESN の計算時間は総じて simpleESN の計算時間より 1 桁小さくすることに成功した。(5-6.7 節)

以上の結果から、ESN モデル、特に計算コストの観点で cycle-groupedESN は自律型電源処理ユニット(SA-PPU)の機能のひとつ、最適作動条件下での電源制御の実現可能性を高める予測モデルあることがわかった。また、 N_x の最適マップやベイズ最適化によって得た知見は、今後の ESN モデル設計を効率的に進める指針になりうると考える。

参考文献

1. 山本直嗣, “連載 えあろすぺーす ABC「ホールスラスト」,” *日本航空宇宙学会誌*, 第 68 巻, 第 4 号, 2020, p. 120.
2. D. M. Goebel and I. Katz, “Fundamentals of Electric Propulsion: Ion and Hall Thrusters,” J. Yuen, Ed. New York: Wiley, 2008, p. 325.
3. “All Electrification Prolongs Lifetime of Satellites,” IHI technical report, p. 45, 2017.
4. 栗木 恭一, 荒川 義博, “電気推進ロケット入門,” 東京大学出版会, 2003.
5. V. Kim, “Main Physical Features and Processes Determining the Performance of Stationary Plasma Thrusters,” *Journal of Propulsion and Power*, vol. 14, no. 5, pp.736-743, 1998.
6. 伊藤寛晟, 山本直嗣, 森野佳生, “リザーバーコンピューティングを用いたホールスラストの電流の経年 変化予想,” 第 61 回航空原動機・宇宙推進講演会 予稿, 鳥取県米子市, 2022.
7. Charles Garner, John Brophy, James Polk, and Lewis Pless, “A 5,730-hr cyclic endurance test of the SPT-100,” AIAA paper 1955-2667, 31st Joint Propulsion Conference and Exhibit, San Diego, 1995.
8. R. R. Hofer, D. M. Goebel, I. G. Mikellides, and I. Katz, “Design of a laboratory Hall thruster with magnetically shielded channel walls, Phase II: Experiments,” in 48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit 2012, Atlanta, 2012.
9. I. G. Mikellides, I. Katz, R. R. Hofer, D. M. Goebel, and S. Member AIAA, “Design of a Laboratory Hall Thruster with Magnetically Shielded Channel Walls, Phase III: Comparison of Theory with Experiment.” 48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit 2012, Atlanta, 2012.
10. I. G. Mikellides, I. Katz, R. R. Hofer, and D. M. Goebel, “Magnetic shielding of a laboratory Hall thruster. I. Theory and validation,” *Journal of Applied Physics*, vol. 115, no. 4, 2014.
11. R. Conversano, “Low-Power Magnetically Shielded Hall Thrusters,” Thesis of doctor degree, California university, 2015.
12. N. P. Brown and M. L. R. Walker, “Review of plasma-induced hall thruster erosion,” *Applied Sciences (Switzerland)*, vol. 10, no. 11. MDPI AG, 2020.
13. E. Sommer, M. K. Scharfe, N. Gascon, M. A. Cappelli, and E. Fernandez, “Simulating plasma-induced hall thruster wall erosion with a two-dimensional hybrid model,” *IEEE Transactions on Plasma Science*, vol. 35, no. 5 I, pp. 1379–1387, 2007.
14. J. T. Yim, M. Keidar, and I. D. Boyd, “An Evaluation of Sources of Erosion in Hall

- Thrusters”. *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Tucson, 2005.
15. M. Gamero-Castaño and I. Katz, “Estimation of Hall Thruster Erosion Using HPHall,” *The 29th International Electric Propulsion Conference*, 2005.
 16. B.A. Arhipov, A.S. Bober, R.Y. Gnizdor, K.N. Kozubsky, A.I. Korakin, N.A. Maslennikov, and S.Y. Pridannikov, “The Results of 7000-Hour SPT-100 Life Testing,” *The 24th International Electric Propulsion Conference*, Moscow, 1995.
 17. T. Meng, C. Qiao, Y. Wang, Z. Ning, and D. Yu, “Accelerated erosion of keeper electrode during coupling discharge between Hall thruster and hollow cathode,” *Vacuum*, vol. 172, 2020.
 18. D. M. Goebel, K. K. Jameson, and R. R. Hofer, “Hall thruster cathode flow impact on coupling voltage and cathode life,” *Journal of Propulsion and Power*, vol. 28, no. 2, 2012, pp. 355–363,
 19. K. G. Xu and M. L. R. Walker, “Effect of external cathode azimuthal position on hall-effect thruster plume and diagnostics,” *Journal of Propulsion and Power*, vol. 30, no. 2, 2014, pp. 506–513.
 20. Y. Matsunaga, H. Watanabe, S. Cho, I. Funaki, H. Kusawake, K. Kajiwarra, F. Kuramoto, and T. Takahashi, “Control algorithm for a 6 kW hall thruster,” *Journal of Electric Propulsion*, vol. 1, no. 1, 2022. p. 29.
 21. M. Kawazu, H. Fuchigami, N. Yamamoto, and T. Tamida, “Neural Network Prediction of Discharge Current using Plume Shape and Operational Parameters in Hall Thrusters,” *THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, vol. 20, 2021, pp. 47–51.
 22. T. Tamida, T. Nakagawa, I. Suga, H. Osuga, T. Ozaki, and K. Matsui, “Determining parameter sets for low-frequency-oscillation-free operation of Hall thruster,” *Journal of Applied Physics*, vol. 102, no. 4, 2007.
 23. 淵上太貴, “ニューラルネットワークによるホールスラスタの制御,” 2017 年度修士論文, 2018.
 24. 河津誠人, “ニューラルネットワークを用いたホールスラスタの自動制御に関する研究,” 2020 年度修士論文, 2021.
 25. H. Jaeger, “The ‘echo state’ approach to analyzing and training recurrent neural networks,” *GMD Report 148, German National Research Center for Information Technology*, 2001.
 26. Y. Pan and J. Wang, “Model predictive control of unknown nonlinear dynamical systems based on recurrent neural networks,” *Institute of Electrical and Electronics Engineers Transactions on Industrial Electronics*, vol. 59, no. 8, 2012, pp. 3089–3101.
 27. T. Sakuma and H. Matsutani, “An area-efficient recurrent neural network core for

- unsupervised time-series anomaly detection,” *The Institute of Electronics, Information and Communication Engineers Transactions on Electronics*, vol. 1, no. 6, 2021, pp. 247–256.
28. F. O. Heimes, “Recurrent neural networks for remaining useful life estimation,” *2008 International Conference on Prognostics and Health Management*, Denver, 2008, pp. 1–6.
 29. S. Hochreiter and J. Schmidhuber, “LONG SHORT-TERM MEMORY,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
 30. 田中剛平, 中根了昌, 廣瀬明, “リザーバーコンピューティング,” 森北出版, 2021.
 31. T. Kailath, A. H. Sayed, and B. Hassibi, “Linear Estimation,” New Jersey: Prentice Hall, 2000.
 32. F.-B. Behrouz, “ADAPTIVE FILTERS: Theory and Applications,” 2nd ed., New York: Wiley, 2013, pp. 416–419.
 33. M. Lukoševičius, “A Practical Guide to Applying Echo State Networks,” *Neural Networks: Tricks of the Trade*, vol. 7700, Berlin: Springer, 2012, pp. 659–686.
 34. M. Lukoševičius and A. Uselis, “Efficient Cross-Validation of Echo State Networks,” *Artificial Neural Networks and Machine Learning – ICANN 2019: Workshop and Special Sessions*, vol. 11731, Berlin: Springer, 2019, pp. 121–133.
 35. C. Gallicchio, A. Micheli, and L. Pedrelli, “Deep reservoir computing: A critical experimental analysis,” *Neurocomputing*, vol. 268, 2017, pp. 87–99.
 36. Z. Li and G. Tanaka, “Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction,” *Neurocomputing*, vol. 467, 2022, pp. 115–129.
 37. K. Zheng, B. Qian, S. Li, Y. Xiao, W. Zhuang, and Q. Ma, “Long-Short Term Echo State Network for Time Series Prediction,” *IEEE Access*, vol. 8, 2020, pp. 91961–91974.
 38. A. E. Hoerl and R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” *Technometrics*, vol. 42, no.1, 2000, pp.80-86.
 39. H. V. Henderson and S. R. Searle, “On Deriving the Inverse of a Sum of Matrices,” *SIAM Review*, vol. 23, no. 1, 1981, pp-53-60.
 40. P. Liashchynskiy and P. Liashchynskiy, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS,” arxiv.org, 2019.
 41. J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” *Advances in Neural Information Processing Systems* 25, vol. 2, 2012. pp. 2951–2959.
 42. 良川太河, 小林学, 後藤正幸, “ベイズ最適化に基づく最適パラメータ探索法の 研究動向と課題に関する一考察,” *情報処理学会第84回全国大会*.2022.
 43. J. Bergstra, D. Yamins, and D. Cox, “Making a Science of Model Search:

- Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures,” *The 30th International Conference on Machine Learning*, vol. 28, no. 1, Atlanta, 2013, pp. 115–123.
44. OptunaTM, Preferred Networks, Inc., <https://www.preferred.jp/ja/projects/optuna/> (Accessed: Jan. 30, 2023)
 45. 長野 公勇, 森田 太智, 山本 直嗣, 窪田 健一, 藤井 剛, 杵淵 紀世志, “200 W 級ホールスラストの開発,” *プラズマ応用科学*, 第 25 巻, 第 2 号, 2017, pp.85-88.
 46. M. Chono, N. Yamamoto, R. Tsukizaki, T. Morishita, K. Kubota, S. Cho, K. Kinefuchi, T. Takahashi, “Performance of a miniature hall thruster and an in-house PPU*,” *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES*, vol. 64, no. 3, 2021, pp. 189–192.
 47. F. Hutter, H. Hoos, and K. Leyton-Brown, “An Efficient Approach for Assessing Hyperparameter Importance,” *The 31st International Conference on Machine Learning*, vol. 32, no. 1, Beijing, 2014, pp. 754–762.
 48. D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *The 3rd International Conference for Learning Representations*, Ithaca, New York, 2015.
 49. Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” *The 33rd International Conference on International Conference on Machine Learning*, vol. 48, New York, 2016, pp. 1050-1059.

謝辞

山本直嗣教授はじめ多くの方のご協力のもと、本研究遂行と修士論文の完成を迎えることが出来ました。山本直嗣先生には、普段から自分のやりたいように研究をさせてもらいつつも、本筋から外れないための多くの助言を承りましたこと感謝致します。森野佳生准教授には、解析手法のアドバイスや査読論文の推敲などで大変お世話になりました、感謝致します。森田太智助教授には普段の研究生活やミーティングを通して、研究に関する新たな気づきや研究者としての心得をたくさん学ばせて頂きました、感謝致します。また、大内田映子秘書、宮本朋子秘書には学会参加に関する手続き等でお世話になり、お二方のおかげで研究活動に集中する事が出来ました。ときには研究生活の悩みの相談相手になって頂いたことも感謝いたします。また、Nathan Schilling 研究員には、アメリカでの面白い話などやジョークで研究室の雰囲気良くして頂きました、感謝致します。また、的確な発表内容に関するアドバイスを頂いた、榊泰直教授と糟谷直宏教授にも感謝致します。

研究生活をしていく中で、多くの助言を頂いた廣池匠哉氏、児島富彦氏、川邊智氏、池邊将暉氏、蔵本英祐氏、児島拓仁氏、松尾涼人氏、胡泽丰氏に感謝致します。また、無事研究生活を終えることができたのも古場健斗氏、金定功樹氏、前之園凱夫氏、手島洸樹氏、村本裕耶氏、小出悠二氏、桂直幹氏、松本遼太郎氏、肥田愛蘭氏のおかげです、感謝致します。また、同期として2年間互いに支え励ましあって研究活動に取り組んだ枝澤友也氏、白木僚氏、稻富泰紀氏、川戸大誠氏、山下裕也氏、植永一汰氏にも感謝致します。最後に、25年間近くで暖かく見守って頂いた両親に感謝の意を表して。謝辞と致します。

付録

A. 各種パラメータの導出

式(1.1)はチャネル内部で電子が $\mathbf{E} \times \mathbf{B}$ ドリフトをする条件である． $\omega_e \tau_e$ は電子が衝突間で旋回運動する回数を意味し，各パラメータは，

$$\tau_e = \frac{1}{\sigma_e n v_e} \quad (\text{A.1})$$

$$\omega_e = \frac{eB}{m_e} \quad (\text{A.2})$$

と表せる．ここで， $\sigma_e [\text{m}^2]$ は電子の衝突断面積， $n [\text{m}^{-3}]$ は中性粒子密度， $v_e [\text{m/s}]$ は電子の速度， e は電気素量， $B [\text{T}]$ は磁場， $m_e [\text{kg}]$ は電子の質量である．また電子の速度 v_e はチャネル内電子を Boltzmann 分布に従うと仮定して次式で表す．

$$v_e = \sqrt{\frac{2k_B T_e}{m_e}} \quad (\text{A.3})$$

式(1.2)は電子を磁場によって加速チャネル内にトラップし，かつイオンが磁場にトラップされることなく加速チャネル外へ放出される条件を表している．電子とイオンのラーマー半径 r_e と r_i はそれぞれ，

$$r_e = \frac{m_e v_e}{eB} \quad (\text{A.4})$$

$$r_i = \frac{m_i v_i}{eB} \quad (\text{A.5})$$

と表す．ここで， $m_e [\text{kg}]$ と $m_i [\text{kg}]$ は電子とイオンの質量， $v_e [\text{m/s}]$ と $v_i [\text{m/s}]$ は電子とイオンの速度である．イオンの速度 v_i は，イオンの熱速度が電場による速度に比べて無視できるほど小さいと仮定した場合，

$$v_i = \sqrt{\frac{2eV_m}{m_i}} \quad (\text{A.6})$$

と表現される．ここで， $V_m [\text{V}]$ はイオンの平均加速電圧である．式(A.1)(A.4)(A.5)(A.6)より，式(1.2)は次式に置き換えることができる．

$$\frac{\sqrt{2m_e k_B T_e}}{e} \ll LB \ll \sqrt{\frac{2m_i V_m}{e}} \quad (\text{A.7})$$

式(1.3)は加速チャネル内でイオンが中性粒子と衝突しないための条件である．イオン中性粒子との衝突断面積を σ_i としたらイオンの平均自由工程 λ_m は次式で表す．

$$\lambda_m = \frac{1}{4\sqrt{2}} \frac{1}{\sigma_i n} \quad (\text{A.8})$$

B. 過学習

過学習は、モデルの表現力が高くなりすぎる起きる現象である。機械学習は、既存の訓練データを用いてモデルを学習させる。多数のノードを用いて、モデル出力と目標出力の誤差が最小になるように学習を進める。このとき、モデルの表現力が高すぎて訓練データに対して当てはまりが良すぎてしまうと、未知のテストデータに対しての予測精度が悪くなってしまう現象が起きてしまう。イメージしやすいのは、多項式フィッティングである。図 B で示すように、本来は2次の多項式で表現するのが正しいが、この次数を高くすると青線のようなフィッティングをしてしまう。これが過学習のわかりやすいイメージである。

この過学習を防ぐために、2-4 節のリッジ回帰では、式(2.15)の右辺第 2 項に正則化項を用意することで、結合重みが大きくなりすぎるのを防ぐことで、過学習が起きないようにしている。

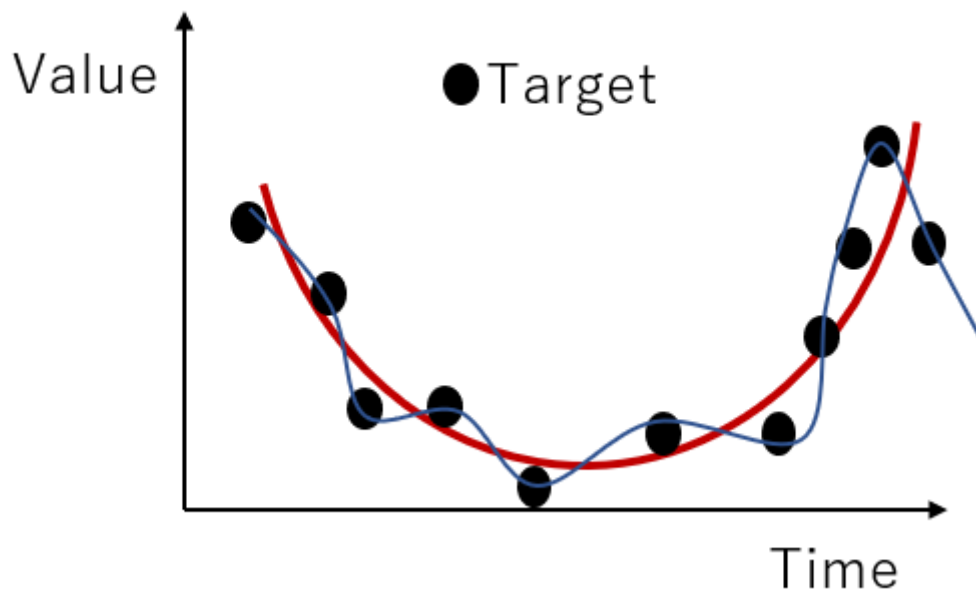


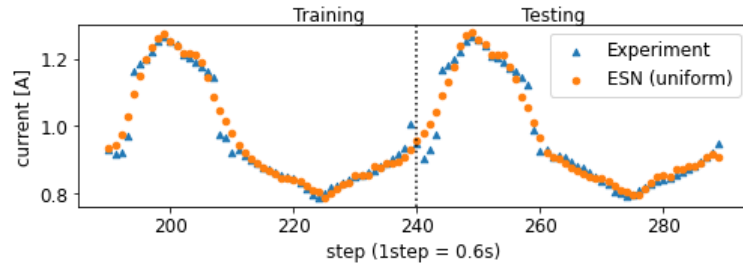
図 B 過学習のイメージ

C. W の要素を決定する分布と予測性能の関係

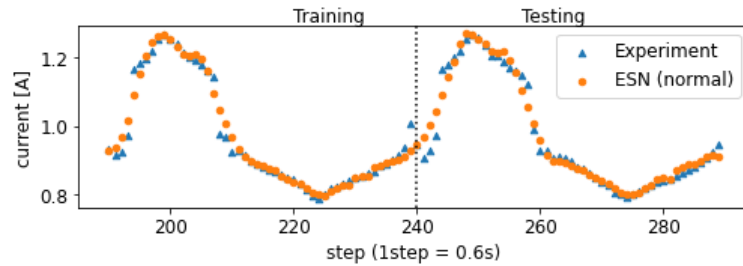
リザーバーのリカレント結合重み行列 W の非ゼロ要素の分布を変えたときの予測精度の違いを確認するために以下の実験を行った。

ESN モデルを用いて放電電流の予測を行った。入力データは 1 ステップ前のコイル電流である。全データは 300 タイムステップあり、前半の 240 ステップを学習データとした。活性化関数は $f = \tanh$ とし、線形学習器はリッジ回帰($\beta = 0.001$)を用いた。ESN の構築に必要な他のハイパーパラメータ(N_x , p , a_{in} , $\rho(W)$, β)はベイズ最適化を 500 試行することで得た。図 C に以下の 3 種類の分布を用いたときの予測結果を示す。3 種類の分布とは、区間 $[-1, 1]$ の一様分布、平均 0、分散1の正規分布、 $\{-1, 1\}$ からランダムに選択する二値分布である。それぞれのテストデータに対する NRMSE は、一様分布は 0.186、正規分布は 0.168、二値分布は 0.184 だった。若干に正規分布の予測精度が高いが、顕著な差は見られず、この優劣もデータやハイパーパラメータの探索方法が変われば、変動すると考える。

(a)



(b)



(c)

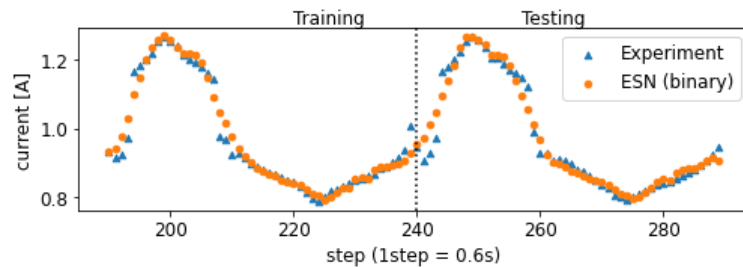


図 C 放電電流の予測結果 (a) 一様分布 (b) 正規分布 (c) 二値分布

D. Cycle-simpleESN

5-1 節で示したサイクル予測を，通常の simpleESN で行った時の結果を示す．5-2 節と同様に，リザーバーのノード数 N_x をスライドさせながら 512 探索のベイズ最適化を用いて，ESN モデルの設計を行っている．図 D に， N_x と NRMSE の関係を示す． $N_x = 50, 60$ 付近で NRMSE の減少が見られたが，それ以外の N_x では NRMSE が非常に大きい． $N_x = 50$ の場合の cycle-simpleNRMSE は 0.0852，表 5-5 の cycle-groupedESN が 0.0912 である． N_x が最適値をとっているときの予測精度は悪くないことがわかるが，図 D のように N_x の最適区間から外れると，急激に予測精度が悪くなるため，今回は予測の安定性の観点から cycle-simpleESN の更なる検討を行っていない．

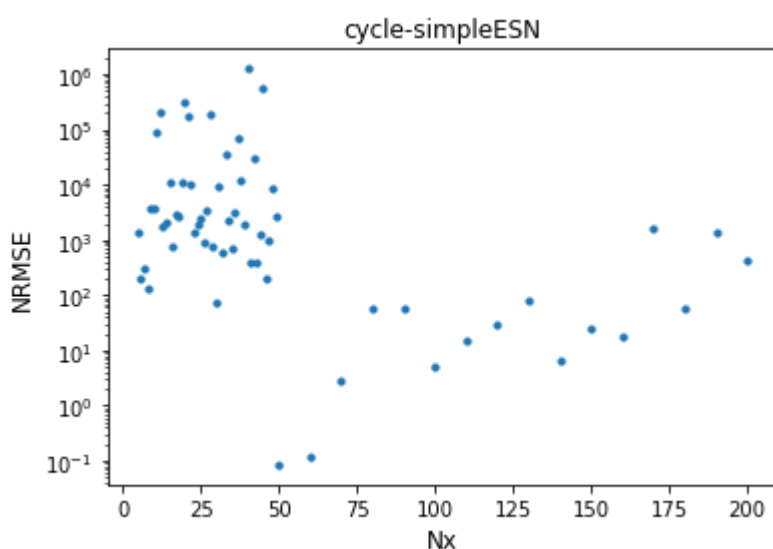


図 D N_x と NRMSE の関係

E. 入力スケールリング a_{in} と非線形性の関係

表 5-1 と表 5-2 で示すように, cycle-groupedESN の最適な入力スケールリング a_{in} は simpleESN の a_{in} よりもはるかに小さい. これは, リザーバー応答がどの程度非線形であるかが, 最適な入力スケールリング a_{in} を決定するためである. 予測タスクが線形に近い場合, 活性化関数の線形に近い部分で式(2.4)のリザーバーの状態ベクトル $\mathbf{x}(n)$ を更新する必要がある. 図 2-3 に示す $\tanh(-)$ の場合, リザーバーの状態は 0 点付近で線形に変換される. 入力結合重み W^{in} のスケールリングを小さくすることで, 0 点付近での変換が可能になる. 一方で, 予測タスクの非線形が強い場合は W^{in} のスケールリングを大きくして, $-1, 1$ 付近の非線形性が高い状態での変換が求められる[33].

予測タスクの非線形性を知るために, simpleESN と cycle-groupedESN のリザーバー 2 を用いて, 入力データとリザーバー状態の関係を調べた. データセット A の放電電流 I_d をモデル出力としている. 両モデルともハイパーパラメータは $a_{in} = 0.1$, $p = 0.1$, $\rho(W) = 0.9$, $\lambda = 0.99$, $\delta = 0.00001$ とした. リザーバーノード数 N_x は simpleESN が 9, cycle-groupedESN のリザーバー層 2 が 70 である. 図 E(a)は simpleESN の全タイムステップの入力 u に対するリザーバー状態ベクトルの任意の成分 x の応答を示している. リザーバー状態ベクトルの成分 x に関しては, 任意の 3 つの成分 x_1, x_2, x_3 を用いている. x の応答が非線形性を持つことがわかる. 図 E(b)は, cycle-groupedESN の入力 u に対する x の応答を示している. サイクル予測はデータ構造を変えたことで, 極めて線形性の高い応答を示しているのがわかる. この非線形性の違いが, 最適な a_{in} のオーダーの違いを生んでいる.

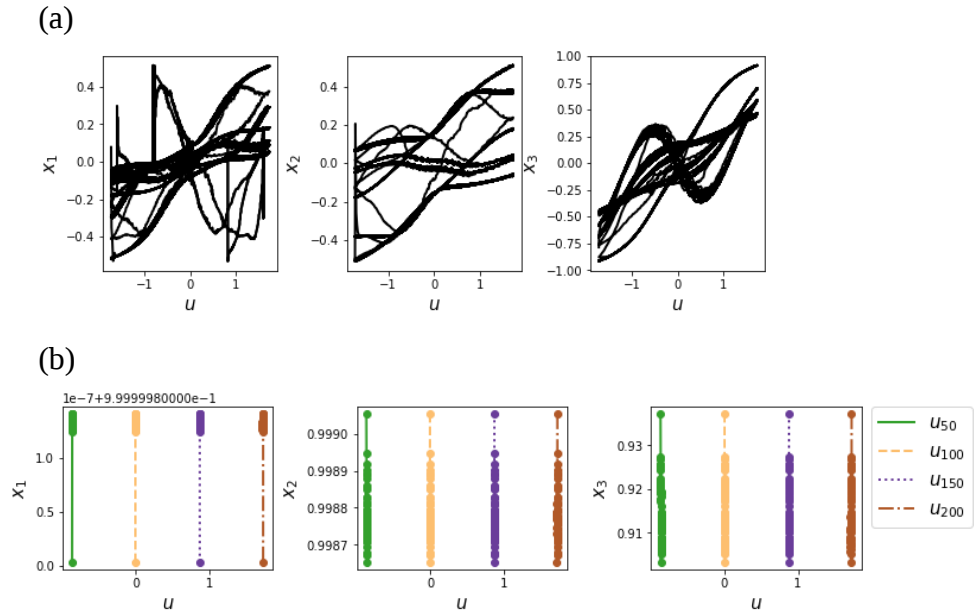


図 E 入力 u に対するリザーバー状態ベクトルの成分 x の応答 (a) simpleESN (b) groupedESN

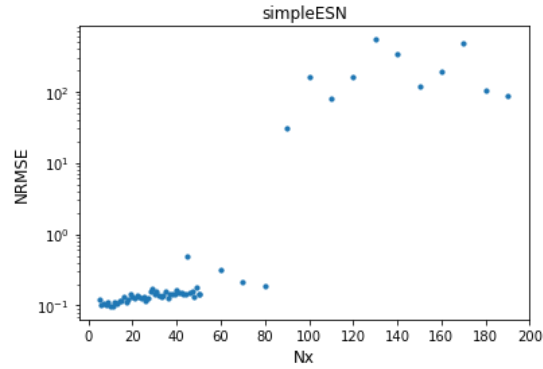
F. 目標出力 V_{cg} のときのノード数 N_x 最適化

5-2 節のリザーバーのノード数 N_x の最適化を目標出力 V_{cg} の場合で行った結果を示す．この最適化を元に，5-6 と 5-7 節の ESN モデルの N_x を決定している．

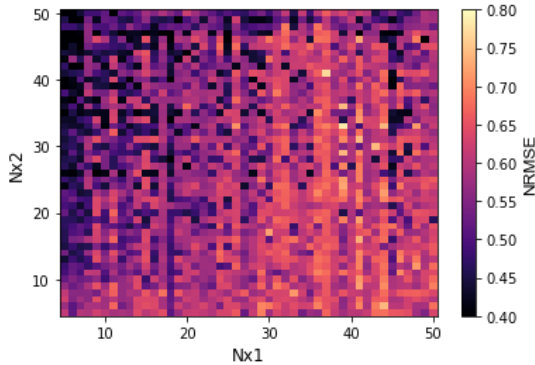
[データセット A]

図 F-1 に， N_x と NRMSE の関係を示す．図 5-2 で示した目標出力が I_d のときと同様，simpleESN は $N_x = 90$ 以降で NRMSE が急激に悪化しているのがわかる．cycle-groupedESN は， $N_{x1} = 5 \sim 8$ 付近で NRMSE が小さいのも， I_d のときと同様である．ただし， $N_{x1} = 150$ 以降で NRMSE が悪化し，0.8 付近の値となっている．

(a)



(b)



(c)

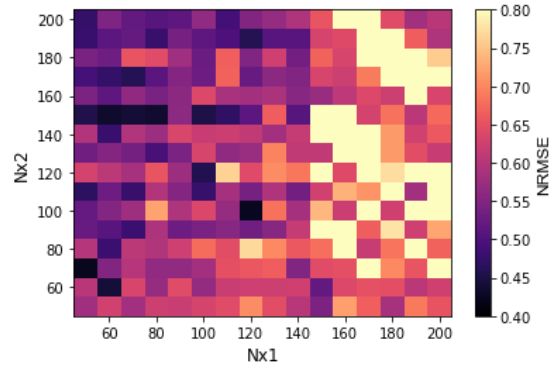
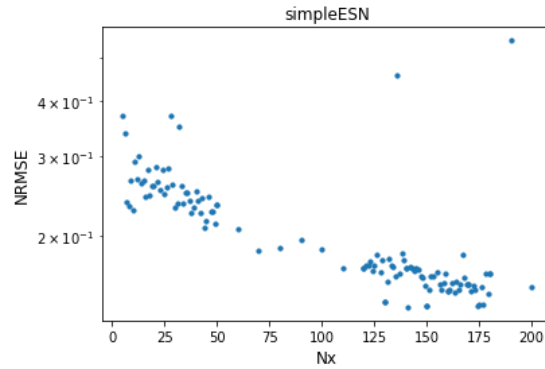


図 F-1 N_x に対する NRMSE マップ (データセット A, 目標出力 V_{cg}) (a) simpleESN (b) groupedESN 区間[5,50] (c) groupedESN 区間[50, 200]

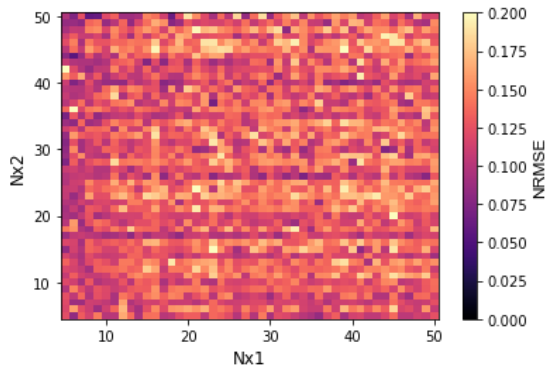
[データセット B]

図 F-2 に、 N_x と NRMSE の関係を示す．図 5-3 で示した目標出力が I_d のときと同様，simpleESN は $N_x = 125$ から 180 付近で NRMSE が小さくなっている．cycle-groupedESN は，マップ全体に NRMSE が小さい領域が広がっている．

(a)



(b)



(c)

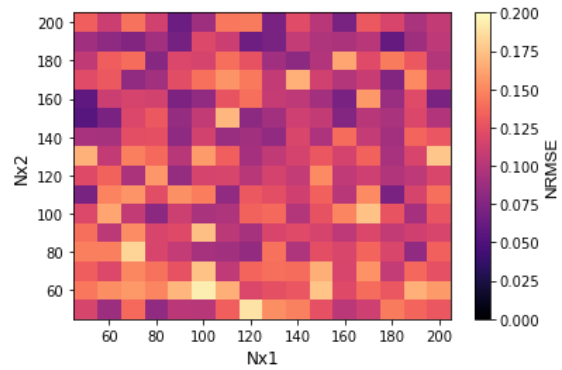


図 F-2 N_x に対する NRMSE マップ (データセット B, 目標出力 V_{cg}) (a) simpleESN (b) cycle-groupedESN 区間[5,50] (c) cycle-groupedESN 区間[50, 200]

G. ハイパーパラメータの分布

5-3 節で 512 試行のハイパーパラメータ探索を 512 探索行った．モデルは simpleESN で， $N_x = 9,60$ と設定している．512 探索分，つまり同一条件の最適化されたハイパーパラメータの結果が 512 個存在する．この結果をヒストグラムで示す．乱数シードを毎探索で変更することで，同一条件でも異なる結果が出力されることになる．図 G にハイパーパラメータの分布を示す．図 G (c) のスペクトル半径 $\rho(W)$ や図 G (e) の RLS 法の正則化項の係数 δ のように，探索空間に幅広く広がっている場合もあれば，図 G (b) の結合確率 p や図 G (d) の忘却係数 λ のように， $N_x = 9,60$ でそれぞれ傾向が異なるハイパーパラメータの分布を得た．各ハイパーパラメータは他のハイパーパラメータと複雑にかかわりあっているため，モデル設計時に一義に定めることは困難である，したがって，ハイパーパラメータ探索によってモデルの予測性能がよくなるハイパーパラメータの組み合わせを探してくる必要がある．

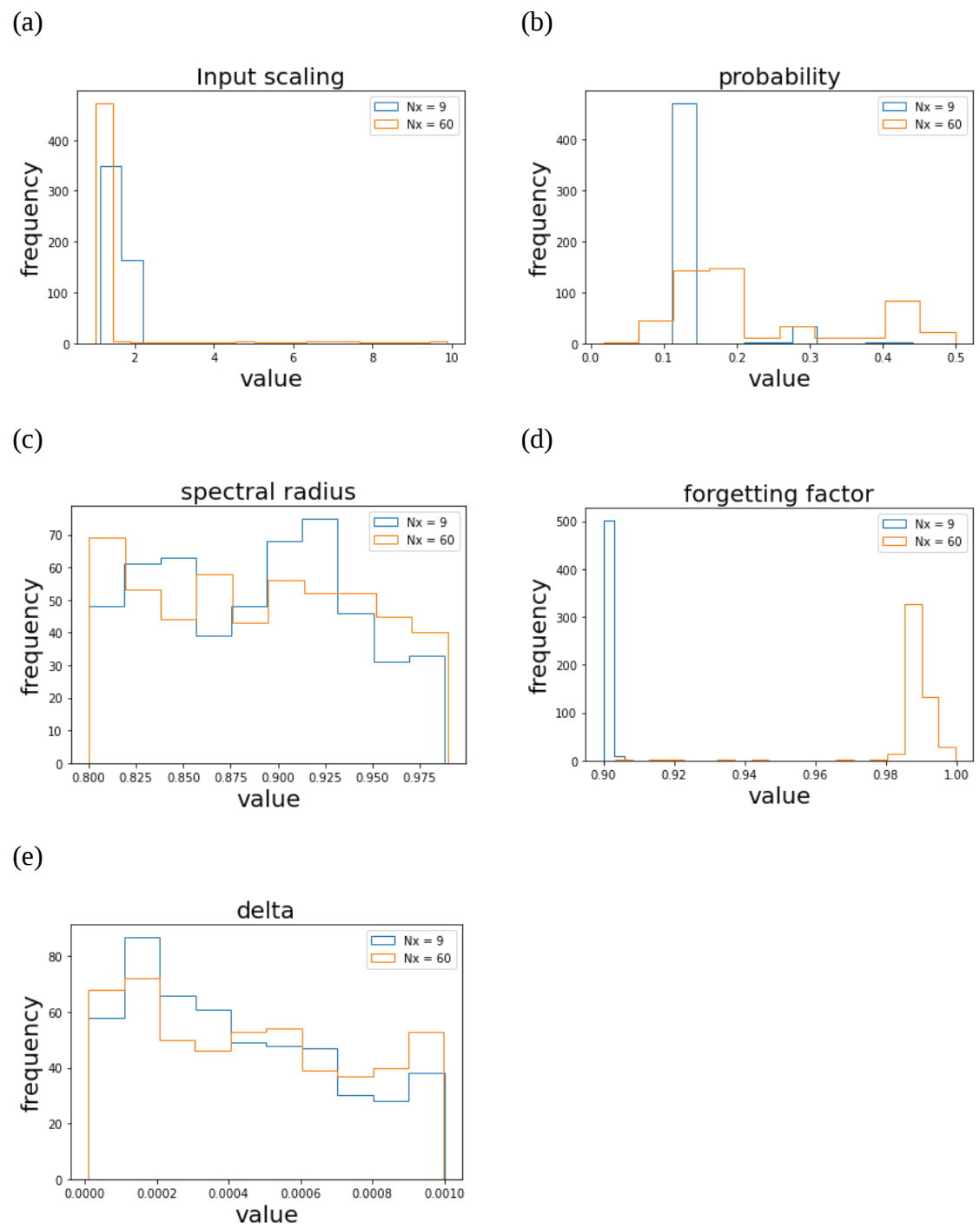
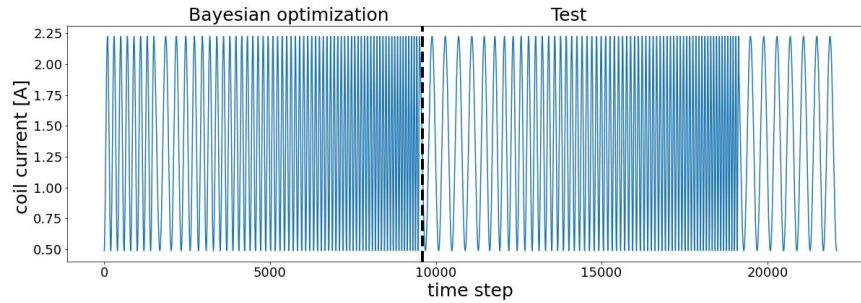


図 G 各ハイパーパラメータの分布 (a) a_{in} (b) p (c) $\rho(W)$ (d) λ (e) δ

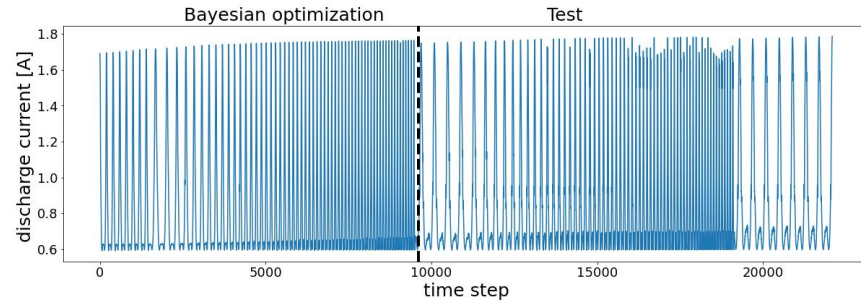
H. データセット C

本文では、データセット A と B の 2 種類を用いた結果を示した．本節では、他のデータセットを用いた場合をまとめる．これをデータセット C と名付ける．4 章の実験から得られたデータセット C は、ローパスフィルタされた信号をサンプリング周波数 2 kHz で取得し、ダウンサンプリングを行ってサンプリング周波数 40 Hz に変換している．放電電圧 300 V、 $\dot{m}$ を 7 sccm に固定している．コイル電流 I_{coil} は図 H-1(a) に示すように、0.5 A と 2.25 A の間を sin 波振動しており、周波数は [0.1 Hz, 0.5 Hz] の間を連続的に変化している．放電電流 I_d を図 H-1(b)，接地電位に対するカソード電位 V_{cg} を図 H-1(c) に示す．総タイムステップ数は 22390 ステップである．

(a)



(b)



(c)

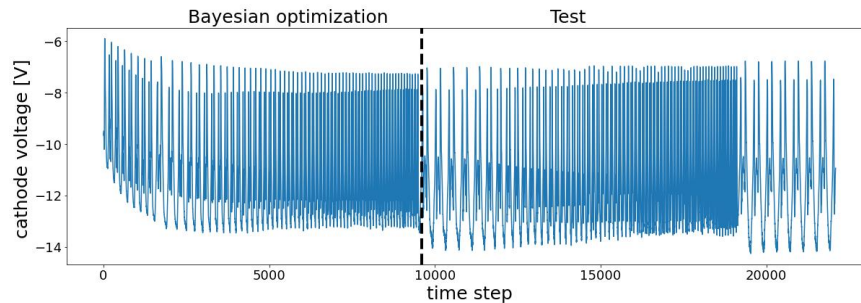
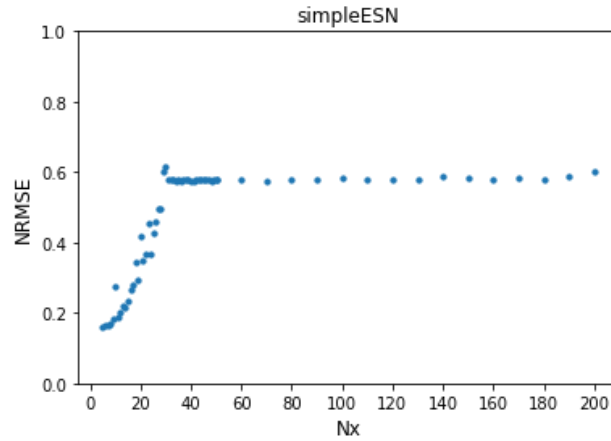


図 H-1 データセット C (a) I_{coil} (b) I_d (c) V_{cg}

5-2 節と同様の方法で、最適な N_x を求めた。データセット C は、コイル電流の周波数が変化するため、1 周期のタイムステップ数が一定ではない。データセット A, B の場合は、1 周期当たり 400 ステップで一定であった。したがって、データセット C は cycle-groupedESN は用いていない。図 H-2 に N_x とテストデータに対する NRMSE を示す。テストデータは 9600 タイムステップ以降のデータである。図 H-2(a) は目標出力が I_d の場合、図 H-2(b) は目標出力が V_{cg} の場合である。どちらの場合も、 N_x の増加とともに NRMSE が増加している。また、 $N_x = 30$ 周辺で NRMSE が頭打ちになり、それ以降 NRMSE = 0.6 付近の一定値をとっている。

(a)



(b)

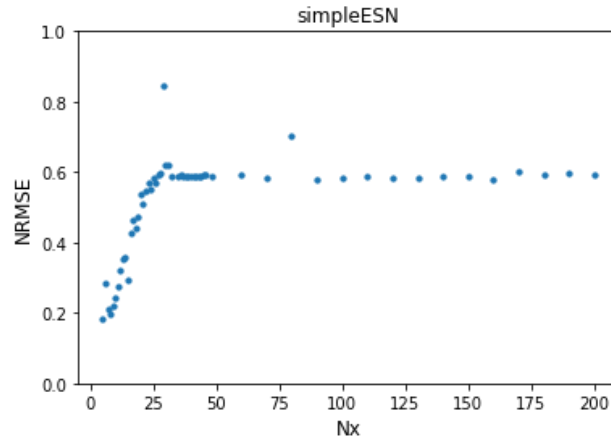


図 H-2 simpleESN の N_x に対する NRMSE マップ (a) 目標出力 I_d (b) V_{cg}

次に，simpleESN モデルと LSTM の予測結果の比較を示す．simpleESN は $N_x = 5, 90$ の 2 パターンのノード数を設定し，LSTM は 5-5 節で示した条件と同一のモデル設計を行った．表 H-1 に目標出力が I_d のとき，表 H-2 に目標出力が V_{cg} のときの NRMSE と計算時間を示す．このとき ESN モデルは 512 試行のベイズ最適化による探索時間も含んでいる． $N_x = 5$ の simpleESN の NRMSE が一番小さく，学習時間も短い．図 H-3 に各モデルのモデル出力と目標出力をプロットする．図 H-3(a) は目標出力 I_d ，図 H-3(b) は目標出力 V_{cg} の結果である．18500 ステップから 21500 ステップまでの結果を示しており，19300 ステップ付近でコイル電流 I_{coil} 周波数の変化が見られる． $N_x = 5$ の simpleESN は当てはまりがよく，周波数の変化にも対応することができている．しかし，図 H-3(a) では， I_d が 0.7A 付近で遷移する区間の予測が上手くいっていない．また，図 H-3(b) では，19500 ステップや 20700 ステップ付近の V_{cg} のピークを予測することが出来ず，モデル出力が大幅に大きいピーク値をとっていることがわかる．

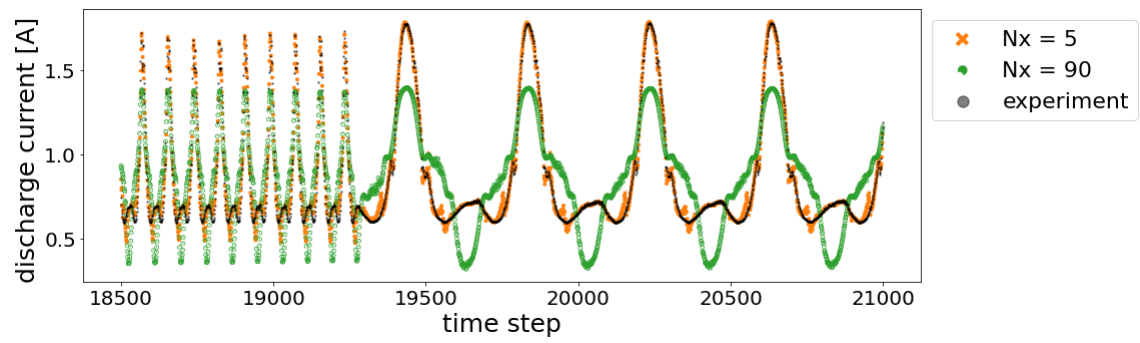
表 H-1 目標出力 I_d のときの予測結果の比較(データセット C)

モデル	N_x	NRMSE	計算時間[秒]
simpleESN	5	0.1403	195
simpleESN	90	0.5957	446
LSTM	-	0.6710	1705

表 H-2 目標出力 V_{cg} のときの予測結果の比較(データセット C)

モデル	N_x	NRMSE	計算時間[秒]
simpleESN	5	0.3608	180
simpleESN	90	0.7288	451
LSTM	-	0.8105	1899

(a)



(b)

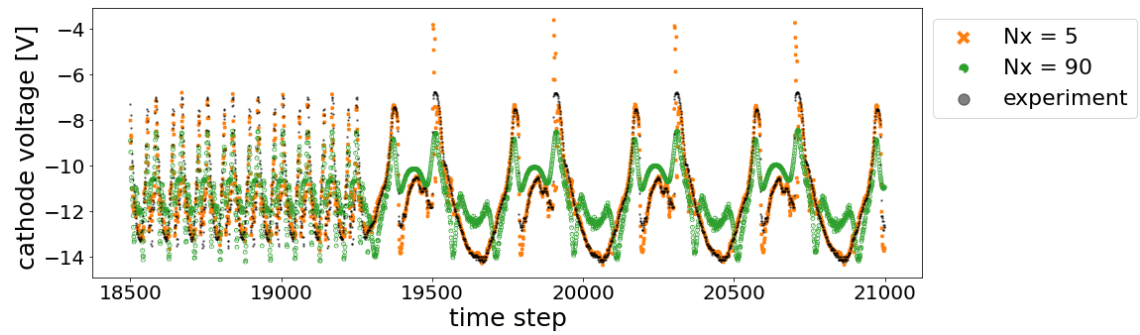


図 H-3 目標出力とモデル出力の比較(データセット C) (a) 目標出力 I_d (b) V_{cg}

I. プルーム画像による放電電流予測

本文で用いた線形学習器は逐次最小二乗法(RLS 法)だったが, 本節ではリッジ回帰(2-4 節)を用いた結果を示す. 実験データは 200W 級ホールスラストを作動させて取得した. ホールスラストの詳細は4-1 節と同様である. 作動条件はコイル電流, 放電電圧, 推進剤流量の 3 パラメータであり, 目標出力は放電電流である. simpleESN モデルを使用し, 入力にはコイル電流と図 I-1 に示すプルーム画像の画素値を用いた. プルームの発色強度や形状が, 放電電流の特性を表現できることが先行研究で示されている[24]. ESN モデルのハイパーパラメータは 500 試行のベイズ最適化により決定している. この予測では, 式(2.5)を,

$$\mathbf{y}(n+1) = \mathbf{W}^{\text{out}}(n)\mathbf{x}(n) \quad (\text{I.1})$$

$$\mathbf{y}(n+10) = \mathbf{W}^{\text{out}}(n)\mathbf{x}(n) \quad (\text{I.2})$$

とすることで, 作動条件とプルーム画像から 1 タイムステップと 10 タイムステップ先を予測するモデルを設計した. 図 I-2 (a)に 1 タイムステップ先予測の結果, 図 I-2 (a)に 1 タイムステップ先予測の結果を示す. テストデータに対する NRMSE は, 1 タイムステップ先予測が 0.147, 10 タイムステップ先予測が 0.230 であった. より遠い将来のステップを予測する場合は予測精度が悪くなる傾向にある.

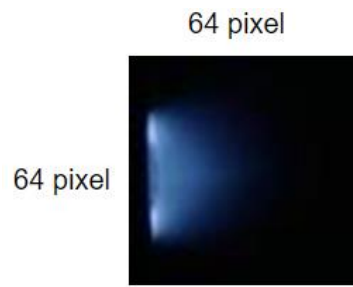
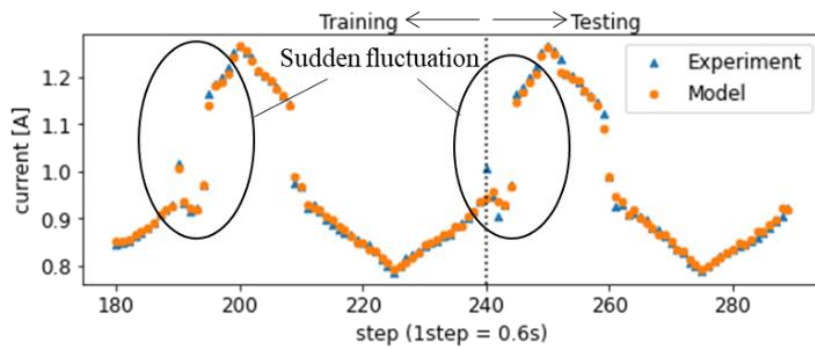


図 I-1 プルーム画像

(a)



(b)

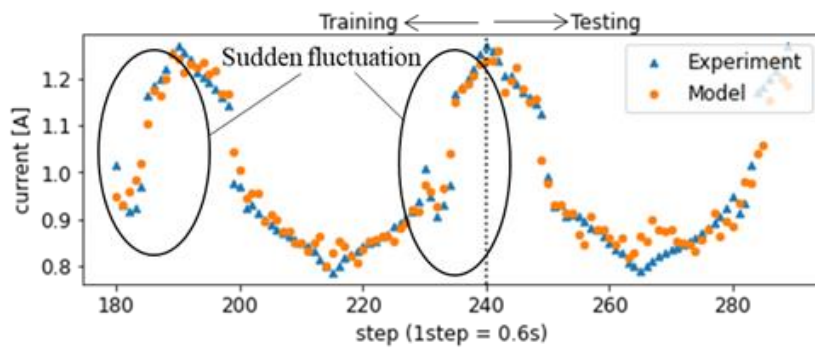


図 I-2 ESN モデルの将来予測の結果

(a) 1 タイムステップ先 (b) 10 タイムステップ先

J. モンテカルロドロップアウトによる不確実性評価

モンテカルロ(MC)ドロップアウトを用いて、モデルの予測性能の不確実性を評価する取り組みも行った。ドロップアウトとは、図 J-1 に示すように、ニューラルネットワークの中間層(リザーバーコンピューティング場合、リザーバー層)のノードをランダム削除する。これは付録 B で示した過学習を抑制する手法として、機械学習の分野ではよく使われる。過学習を抑制するためにドロップアウトを用いる場合は、訓練時のみにドロップアウトを適用する。このドロップアウトを推論時にも適用することでモデルの不確実性を評価する手法が MC ドロップアウトである[49]。これを ESN モデルにも適用することで、不確実性を評価する。ランダムにノードを消去することを複数回行う事で、構造が若干異なるリザーバー層が複数生成される。そのモデルを用いて通常の予測タスクを行う。本研究ではドロップアウト率を 1% としてドロップアウトを 100 回行い、その分布を得ている。分布からモデル出力の平均値と両側 95%予測区間をノンパラメトリックな方法で算出した。両側 95%予測区間とはモデル出力(予測結果) $y(n)$ が両側 95 %の確率でこの区間に入っていることを表す。用いた実験データは 200W 級ホールスラストを作動させて得た。simpleESN モデルへの入力はコイル電流のみであり、目標出力は放電電流である。simpleESN の線形学習器は RLS であり、ハイパーパラメータは $N_x = 100$, $p = 0.3$, $a_{in} = 10$, $\rho(W) = 0.9$, $\delta = 0.4$, $\lambda = 0.001$ としている。図 J-2 に両側 95%予測区間も含めた結果を示す。モデルの不確実性を評価するひとつの指標として MC ドロップアウトは有効である。また、このデータの場合も、タイムステップが増えるにつれて当てはまりが良くなるオンライン学習の特徴を確認できた。

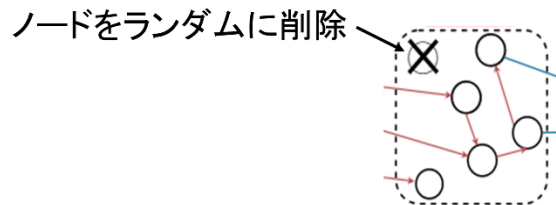


図 J-1 ドロップアウトのイメージ

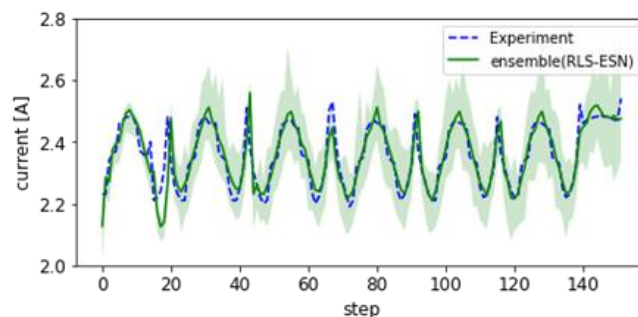


図 J-2 MC ドロップアウトを用いたときの予測

K. 忘却係数 λ の最適化

本文では、表 5-1 と表 5-2 に示すハイパーパラメータはリザーバーのノード数 N_x を除いて、ベイズ最適化を用いて最適化している。しかしながら、式(2.41)に示すように、RLS アルゴリズムのコスト関数の正則化項（第 2 項）は、係数に忘却係数 λ と正則化パラメータ δ が存在する。これは、この 2 つのハイパーパラメータが、正則化項として同一の役割を果たしていることを意味する。本節では、 λ と δ を独立に最適化することを考える。具体的には、予め λ と N_x を固定し、 δ を含む他のハイパーパラメータを 512 試行のベイズ最適化で用いて決定する。 λ は 0.7 から 1 の間をスライドさせていく。0.70 から 0.90 の区間は刻み幅 0.01 とし、0.900 から 1.000 の区間は刻み幅を 0.001 に設定する。データセット A を用いて、simpleESN と cycle-groupedESN の放電電流 I_d 予測タスクを考える。 N_x は表 5-6 より、simpleESN は $N_x = 9$ 、cycle-groupedESN は $(N_{x1}, N_{x2}) = (6, 31)$ を選択している。

図 K.1 に各モデルでの λ に対するテストデータの NRMSE を示す。simpleESN の方は、忘却係数 $\lambda = 0.9$ 付近で NRMSE が小さくなっているため、最適な λ は 0.9 付近だといえる。 λ を 0.9 より大きくしていくと、NRMSE は増加に転じているのか確認できる。一方で、cycle-groupedESN は、 λ が変化しても NRMSE に変化は見られない。

次に、忘却係数 λ と最適化された正則化パラメータ δ の関係を図 K-2 に示す。simpleESN1 は、探索空間内でばらつきが大きいため、最適な正則化パラメータ δ を一義に決定する事は困難である。一方で、cycle-groupedESN は $\lambda = 0.7 \sim 0.85$ 付近ではおよそ $\delta = 6.7 \times 10^{-4}$ の一定値をとっており $\lambda = 0.9$ 前後で δ の値が急激に小さくなる現象が見られた。

忘却係数を最適化することは、RLS アルゴリズムのコスト関数の改良につながる。式(2.41)のコスト関数は誤差の二乗和であり、 λ で重みづけされた各項は過去遡るにつれて、二乗和への寄与が小さくなる。したがって、限りなく 0 に近い項は無視してもいいはずである。言い換えれば、コスト関数の計算の際に、ある程度過去のデータを捨てることができる。これは、計算資源が限られるオンボードコンピューティングのメモリの節約につながる。例えば、simpleESN モデルの忘却係数 λ を 0.9 と設定した場合、1 サイクル前(400 ステップ前)の誤差項にかかる係数は $0.9^{400} = 5 \times 10^{-19}$ となり、無視できるほど小さいと考える。ただし、データを捨てても、捨てない場合と同じように予測できるとは限らないので、RLS アルゴリズムの改良も含めて、更なる検討が必要である。

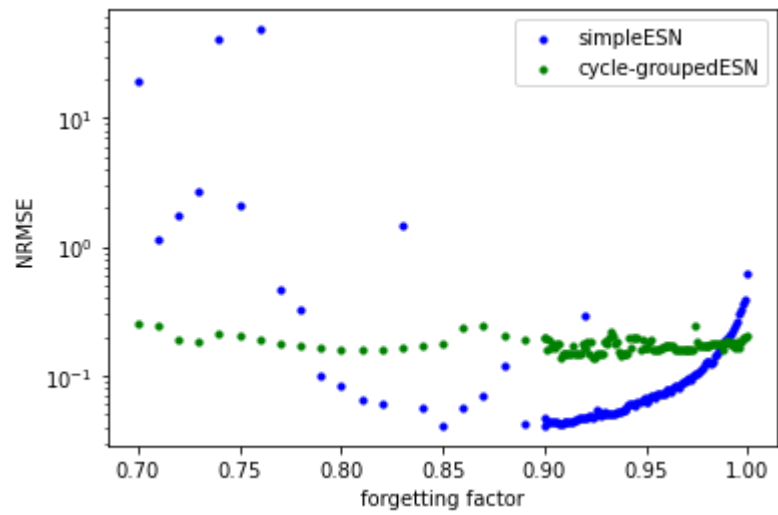


図 K-1 忘却係数 λ と NRMSE の関係

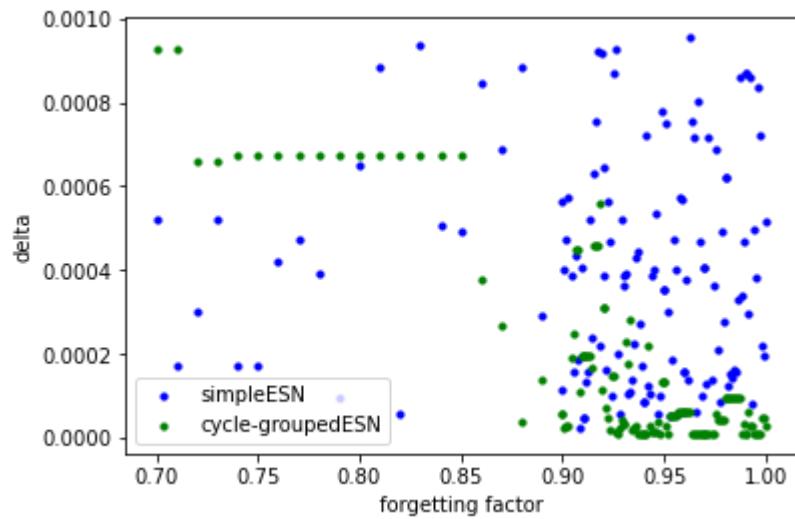


図 K-2 忘却係数 λ と正則化パラメータ δ の関係

L. 出力結合重み W^{out} の時間変化

出力結合重み行列 $W^{out} \in \mathbb{R}^{N_d \times N_x}$ の複数の成分 w^{out} の時間変化を確認する事で、本データセットにおける w^{out} の揺らぎを調査した。データセット A を用いて、simpleESN と cycle-groupedESN の 2 つのモデルで調査した。目標出力は放電電流 I_d と接地電位に対するカソード電位 V_{cg} の 2 種類である。simpleESN は行列 W^{out} のすべての成分、cycle-groupedESN は W^{out} から適当に (N_{x1}, N_{x2}) 成分を選択し、図 L-1, 図 L-2 に示す。図 L-1(a)と(b)は simpleESN で I_d を予測したときの W^{out} の成分 w^{out} の時間変化である。図 L-1(a)は $N_x = 9$, 図 L-1(b)は $N_x = 60$ のときを示している。図 L-1(a)からわかる通り、 W^{out} の成分は定常状態を保っているものの、急激に増加(減少)する区間が見られる。 $|w^{out}|$ が急激に大きくなる区間は、放電電流 I_d が減少している区間と合致している。また、この区間の当てはまりは I_d が増加しているときより悪い。一方で、図 L-1(b)で示すように、 $N_x = 60$ の場合は急激な揺らぎは見られない。図 L-1(c)と(d)に cycle-groupedESN で I_d を予測したときの W^{out} の成分の時間変化を示す。グレーの点線は、放電電圧 V_{cg} が切り替わる点を示しており、青の点線はベイズ最適化に用いた区間とテストデータに用いた区間の境目である。図からわかる通り、同一の V_{cg} 内では w^{out} のばらつきは小さく、 V_{cg} が切り替わるタイミングが増えるにつれて $|w^{out}|$ が大きくなっていることがわかる。図 L-1(c)と(d)を比較すると、 $(N_{x1}, N_{x2}) = (50, 70)$ のときのほうが w^{out} の増加が抑えられている。

同様に、図 L-2(a)と(b)に simpleESN で V_{cg} を予測した場合、図 L-2(c)と(d)に cycle-groupedESN で V_{cg} を予測した場合の w^{out} の変化を示す。図 L-1 と同様の傾向が見られる。以上の結果より、simpleESN においては W^{out} の成分 w^{out} が急激に変化することを抑制する方法をとることで更なる予測精度の向上が見込める。また、リザーバーのノード数 N_x を大きくすると w^{out} が急激に変化することを抑制できることがわかった。リザーバーのノード数 N_x が小さくなる理由や、Cycle-groupedESN において作動条件が変わるごとに $|w^{out}|$ が増加する原因は未解明のため更なる調査が必要である。

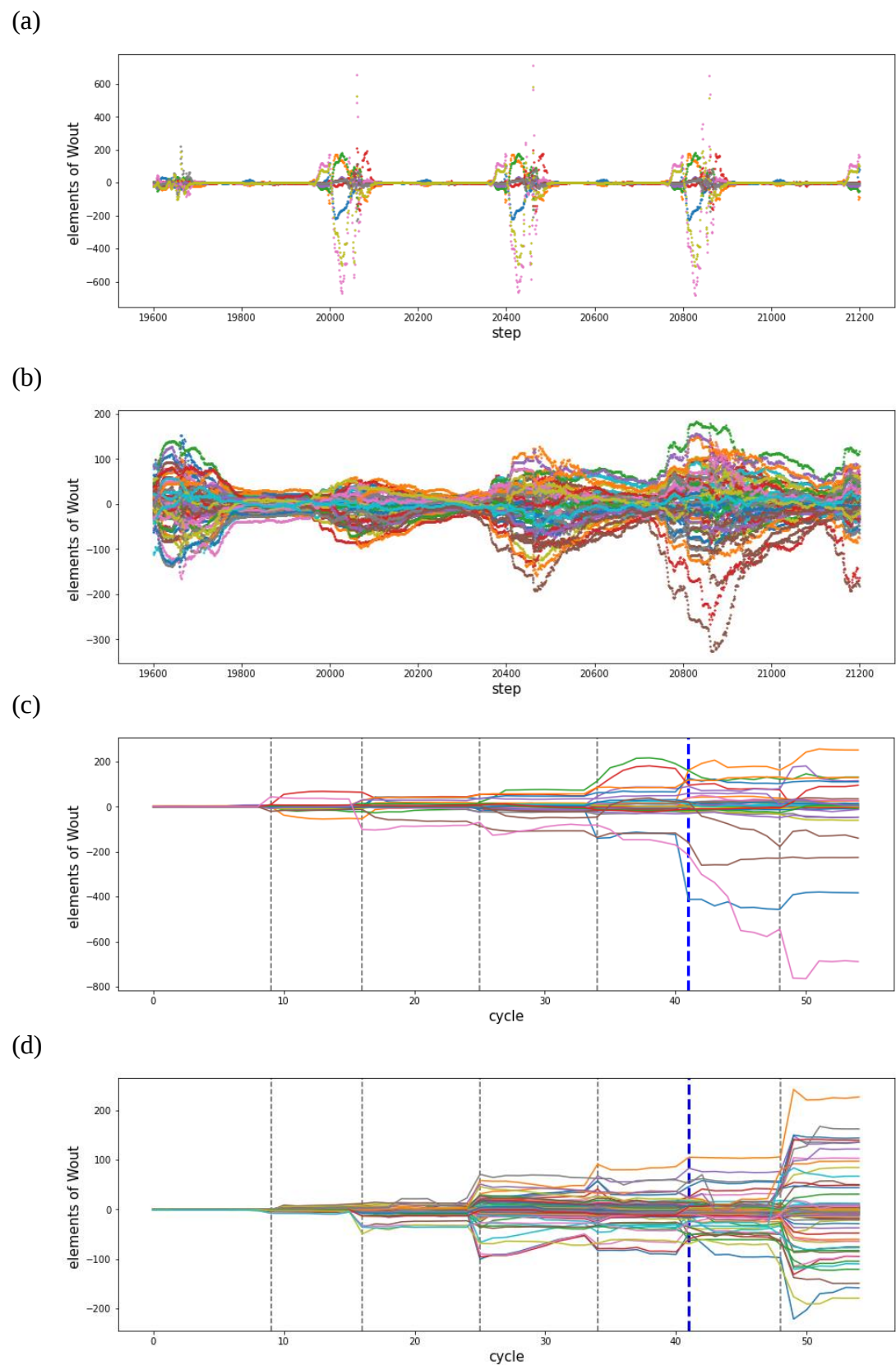
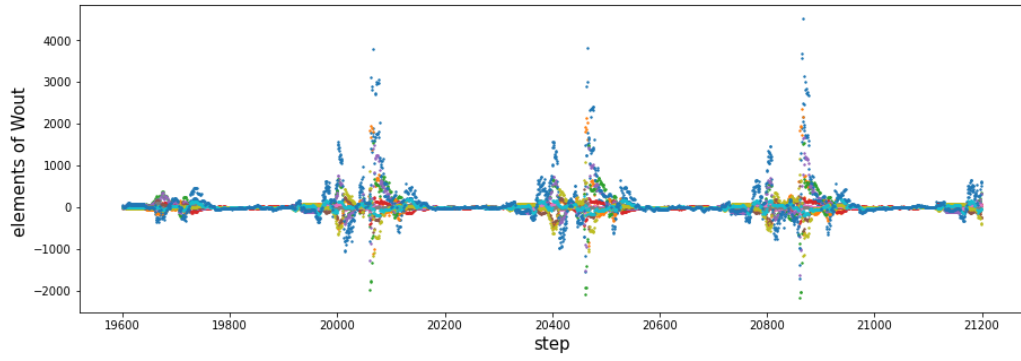
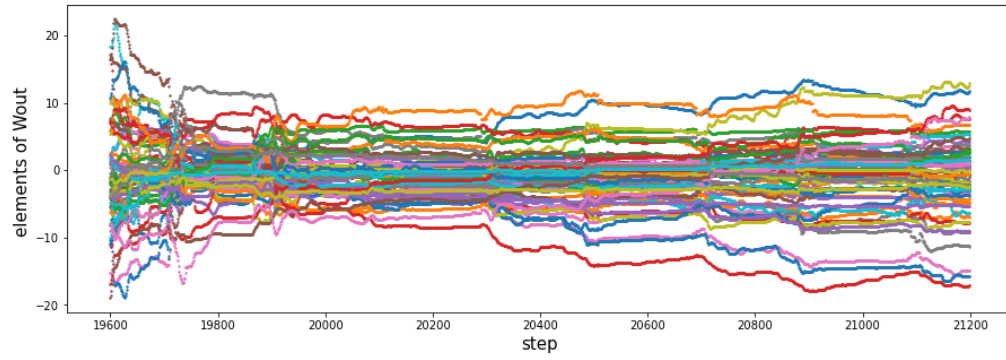


図 L-1 目標出力 I_d の場合の、 w^{out} の時間変化 (a) simpleESN ($N_x = 9$) (b) simpleESN ($N_x = 60$) (c) cycle-groupedESN ($(N_{x1}, N_{x2}) = (6, 31)$) (d) cycle-groupedESN ($(N_{x1}, N_{x2}) = (50, 70)$)

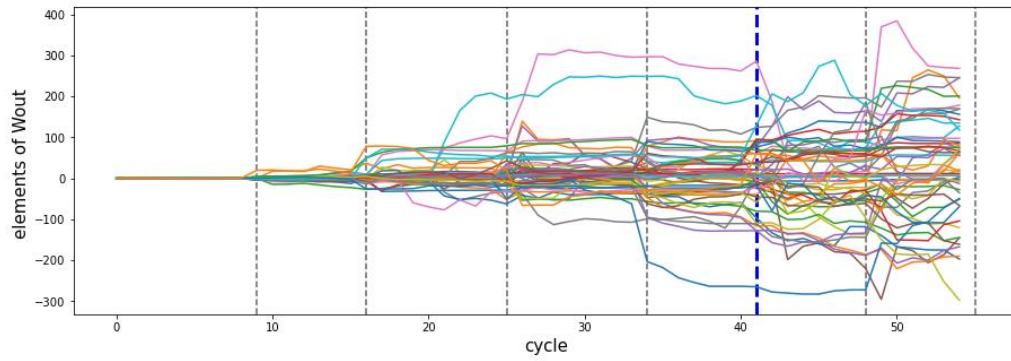
(a)



(b)



(c)



(d)

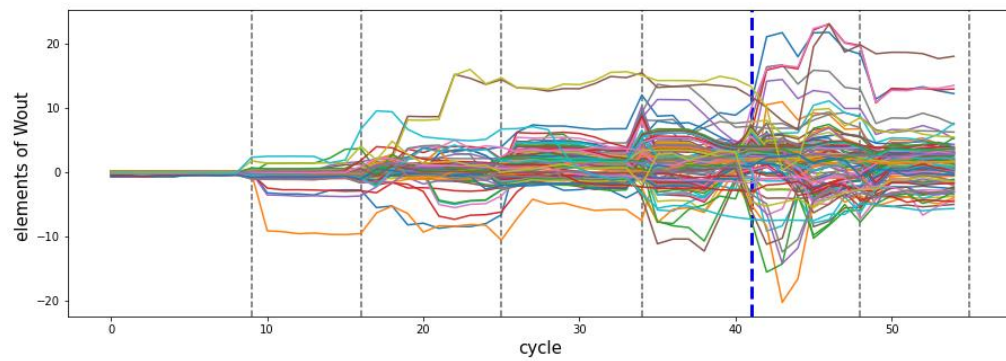


図 L-2 目標出力 V_{cg} の場合の, w^{out} の時間変化 (a) simpleESN ($N_x = 11$) (b) simpleESN ($N_x = 60$) (c) cycle-groupedESN ($(N_{x1}, N_{x2}) = (7, 45)$) (d) cycle-groupedESN ($(N_{x1}, N_{x2}) = (50, 70)$)