

機械学習と CAE を利用したターボ機械の設計支援技術

Design Support Techniques for Turbocharger using Machine Learning and CAE

齊 藤 弘 樹 技術開発本部基盤技術研究所数理工学研究所
服 部 均 技術開発本部基盤技術研究所数理工学研究所
米 倉 一 男 技術開発本部基盤技術研究所数理工学研究所 博士（情報理工学）

機械学習と CAE による設計支援を目的とし、性能予測として Deep Neural Network による圧縮機のサージ流量予測、設計解探索として Deep Q-Network による LPT 翼の圧力損失最小化を行った。その結果、性能予測では設計上流で用いられる一次元物理モデルと比較して高精度な予測ができることが確認できた。Deep Neural Network は物理モデルを用いない一方で高精度な予測が可能であり、未解明な物理現象への適用が期待できる。また、設計解探索では幅広い条件で学習することにより汎用的に探索が可能なること、設計解の根拠を可視化することにより設計者へのフィードバックが可能であることが確認できた。

Recently CAE is becoming widely used in the design of turbo machinery, such as turbochargers and jet engines, to predict its performance. In some specific cases, however, it is difficult to predict the performance using CAE. This is mainly because the physical phenomena are not completely clarified due to their complexities. Furthermore, CAE is also used in combination with the optimizing techniques, such as genetic algorithms and response surface methods, to obtain better design solutions. Although many simulations are performed and accumulated, these optimization techniques cannot utilize the accumulated knowledge. In this research, two machine learning techniques, deep neural network and deep Q-network, are applied to two problems respectively: performance prediction of the volumetric flow at the surge of compressors and a shape optimization targeting minimization of pressure drop of the low-pressure turbine airfoils. The results show that deep neural network has high predictive accuracy while not requiring physical models. It has also been confirmed that deep Q-network acquires high generalization capabilities to be applied to different scenarios by training with various conditions. In addition, it has been revealed that deep Q-network puts large weight at the same point as well-trained designers do.

1. 緒 言

近年、過給機や航空用エンジンなどの流体機械の設計に CAE (Computer Aided Engineering) が広く使われるようになってきた。CAE を利用することで製品を解析可能な範囲でモデル化し、数値解析的に性能を予測することが可能になる。しかしながら、CAE だけでは性能が十分に予測できない場合があり、圧縮機のサージ現象がその例として挙げられる。

車両用過給機では圧縮機を含む管路系全体において、サージと呼ばれる大振幅の圧力変動および流量変動を伴う自励振動現象が発生すると、その役割を果たせなくなるだけでなく、圧力変動や振動によって圧縮機やその周辺機器が損傷する場合がある。このため、サージは発生回避が不可欠な現象であり、圧縮機の作動域拡大のためには設計段階で圧力比ピーク点を予測することが重要である。サージ流量（圧力比ピーク点の体積流量と定義）を予測するためには、複数の作動点で三次元定常 CFD (Computational

Fluid Dynamics) 解析を実施し、圧縮機全体の性能すなわち圧力比-流量特性を予測する手法が一般的である。しかしながら、基本設計段階では三次元形状が決定しておらず、想定し得る全ての形状に対して CFD 解析を実施することは計算コストの観点から現実的ではない。また、三次元形状が決定したとしても、サージ流量付近の小流量側は、旋回失速などの非定常現象によって性能が低下する領域であり、計算も不安定となるため、定常解析では圧縮機全体の性能を見誤ることが多い。非定常解析を実施したとしてもサージ流量を定量的に再現することは困難であり、さらにはサージ流量付近の小流量側における流れ場の理解が不十分である。これらの理由から、設計段階で CFD を用いてサージ流量を予測する手法は確立されておらず、過去の類似する形状から定性的に類推するなど、設計者の知見に依存しているのが現状である。このため、設計の後戻りを招き、製品開発コストの増大につながっている。サージ現象を解明し、物理モデル化することが予測手法確立のために重要であるが、現時点ではまだ多くの研究が必要で

ある。このような場合、現象が解明されている部分は物理モデルで予測し、それ以外の未解明部分はデータに基づく機械学習で補完するシステムが有効であると考えられる。このシステムは、現象の解明が進むに従って、機械学習の部分を物理モデルで置き換えることで、より信頼性が高いものとなる。

また、CAEの活用はより良い設計解を得ることが目的であり、多くの数値解析を繰り返すことで設計解の探索が行われる。これらの多数の試行と解析結果は大きな資産であり、これを分析、活用することでより良い設計解が短時間で得られる可能性がある。これまでに遺伝的アルゴリズム⁽¹⁾や応答曲面法⁽²⁾を用いた最適化が行われてきたが、これらの手法は、流れの境界条件が異なり過去の解析結果を利用できない場合には一からやり直す必要があり、過去の解析結果を資産として活用することが困難である。しかしながら、例えばベテラン設計者であれば、新しい設計条件であっても過去の設計の経験からある程度の推論を行うことができるように、流れの境界条件が変わっても、過去の解析の知見は何らかの方法で利用できるはずである。このような能力を機械学習により実現しようという研究⁽³⁾があり、特定の条件だけでなく汎用的に使えるという意味で汎化性能といわれる。本稿で扱う強化学習は、機械学習のなかでは比較的新しい技術であるが、汎化性能が高いことが特長である。この汎化性能の高さを設計に適用できれば、従来の最適化手法で必要だった解析時間を短縮できる可能性がある。さらに、機械学習が学習した設計のポイントを設計者にフィードバックすることで、機械学習モデルの健全性を検証するとともに、設計者に新たな気づきを与えられる可能性もある。このようなフィードバックの可能性についても本稿で検討する。

以上のように、本稿では物理モデルの未解明部分を補完するためのDNN(Deep Neural Network: 深層ニューラルネットワーク)⁽⁴⁾を用いたサージ流量予測手法、強化学習の一つであるDQN(Deep Q-Network: 深層Q学習)⁽⁵⁾の汎化性能と過去の解析データを活用した設計解探索手法について述べる。

2. 機械学習とCAEを用いた設計支援技術

2.1 機械学習技術概要

2.1.1 Deep Neural Network

近年、データ間の法則性や特徴を見だし、未知の値を予測する手法として機械学習が注目されており、NN

(Neural Network: ニューラルネットワーク)はその代表的な手法の一つである。NNは複数のノードを層状に結合し、学習によって重みと呼ばれるノード間の結合強度を変化させ、複雑な非線形関数を表現できる。そのため、サージのような複雑な現象であっても、豊富なデータから形状パラメータとサージ流量を関係付け、サージ流量を予測できると考えられる。機械製品にNNを適用した例として、エンジンの効率とNO_xなどの汚染物質排出量の予測と最適化⁽⁶⁾、⁽⁷⁾、ディーゼルエンジンの失火予測⁽⁸⁾やリラクタンスモータの回転子の位置予測⁽⁹⁾などが挙げられる。また、スラグやフライアッシュなどの量からコンクリートの圧縮強度を予測した例も報告されている⁽¹⁰⁾。

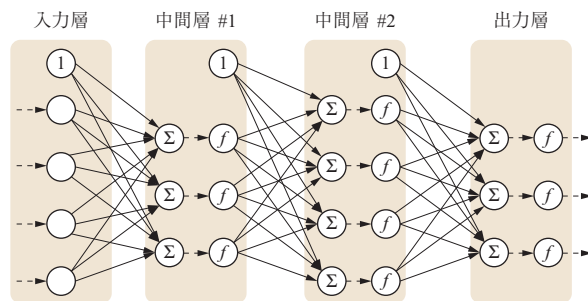
第1図は4層NNの構造を示したものであり、入力層と出力層の間に中間層が2層ある。第1図のNNは各層間のノードが全て結合されており、この構造のNNをFCNN(Fully-Connected Neural Networks: 全結合ニューラルネットワーク)と呼ぶ。NNは受け取った値をそのまま次へ渡す入力ノード、常に1を渡すバイアスノード、受け取った値の総和を次へ渡す総和ノード、受け取った値を関数に通して次へ渡す活性化関数ノードで構成される。総和ノードと活性化関数ノードでは以下の計算が行われ、次の層へ出力を渡す。

$$o_{i,\text{sum}}^l = b_i^l + w_{i1}^l o_{1,\text{func}}^{l-1} + w_{i2}^l o_{2,\text{func}}^{l-1} + \dots + w_{iN^{l-1}}^l o_{N^{l-1},\text{func}}^{l-1}$$

$$= b_i^l + \sum_{k=1}^{N^{l-1}} w_{ik}^l o_{k,\text{func}}^{l-1} \quad \dots \dots \dots (1)$$

$$o_{i,\text{func}}^l = f(o_{i,\text{sum}}^l) \quad \dots \dots \dots (2)$$

ここで、 $o_{i,\text{sum}}^l$ と $o_{i,\text{func}}^l$ はそれぞれ入力層を $l=0$ とした第 l 層のノード i の総和ノードと活性化関数ノードの出力



- (注) ○ : 入力ノード
 ① : バイアスノード
 Σ : 総和ノード
 f : 活性化関数ノード
 → : 重み, バイアスを掛ける
 --> : 値をそのまま渡す

第1図 NNの構造
 Fig. 1 Architecture of NN

力, b_l^i はバイアス, w_{ik}^l は第 $l-1$ 層の k 番目のノードから第 l 層のノード i に結合する重み, N^{l-1} は第 $l-1$ 層のノードの数を表す.

$f(\cdot)$ は活性化関数と呼ばれ, Sigmoid, tanh, ReLU⁽¹¹⁾, leaky ReLU⁽¹²⁾ などの非線形関数がよく使われる. 入力層から出力層までの全てのノードで (1) 式と (2) 式の計算を繰り返して最終的な出力を得るため, 一般的に層数とノード数は多い方が NN は複雑な関数を表現できるようになり, 中間層が 2 層以上のものを特に DNN と呼ぶ.

NN の学習のなかでも, 入力 x とそれに対応する答え t を与え, NN の出力 y を答え t に近づけるように重み w とバイアス b を更新する学習を教師あり学習と呼ぶ. 教師あり学習で行うタスクは回帰と分類であり, 本稿では過去の試験データに基づいてサージ流量の値を予測するため, 教師あり学習の回帰を行う. 回帰における重みの更新は以下の式を用いる.

$$w_{ik}^l \leftarrow w_{ik}^l - \eta \frac{\partial E}{\partial w_{ik}^l} \quad \dots\dots\dots (3)$$

ここで, η は学習係数, E は誤差関数であり, 回帰では以下の二乗誤差を用いるのが一般的である.

$$E = \frac{1}{2} \sum_{k=1}^{N^L} (t_k - o_{k, \text{func}}^L)^2 + \frac{1}{2} \lambda \sum_{l=1}^L \sum_{i=1}^{N^l} \sum_{k=1}^{N^{l-1}} (w_{ik}^l)^2 \quad \dots\dots\dots (4)$$

N^L は出力層のノード数, $o_{k, \text{func}}^L$ は出力層の出力 ($=y_k$), t_k は対応する答えである. また, 第 2 項は後述する過学習を抑制するための正則化項であり, λ はその抑制の強さを表す正則化率である. (3) 式の学習係数 η は学習速度や局所解に陥るかどうかに関係し, 学習に伴って最適な値に変化させるさまざまなアルゴリズムが提案されている. そのなかでも Adam⁽¹³⁾ は安定して高精度な学習結果が得られるアルゴリズムであり, 広く使われている.

NN に限らず, 機械学習による教師あり学習では, 学習に用いたデータに対して過剰に適合し, 未知データに対する予測性能が低くなる過学習と呼ばれる現象が問題になることがある. (4) 式の第 2 項の正則化は過学習を抑制するために導入されているが, そのほかにドロップアウト⁽¹⁴⁾ が広く用いられる. ドロップアウトは学習時にある一定の割合で中間層のノードを学習から除外する手法である. 学習用データが NN に与えられるたびに除外するノードをランダムに変えることで, 複数の NN で学習した結果の平均を取るような効果が得られ, 過学習の抑制に効果的に機能する.

2.1.2 Deep Q-Network

RL (Reinforcement Learning: 強化学習) は, 特定の環境下で設定された価値を最大化する行動を試行錯誤的に学習する機械学習の一つである. RL が使われた例として, プロの囲碁棋士に勝利した AlphaGo⁽¹⁵⁾ やロボットアーム⁽¹⁶⁾, 自動車の自動運転⁽¹⁷⁾ が挙げられる.

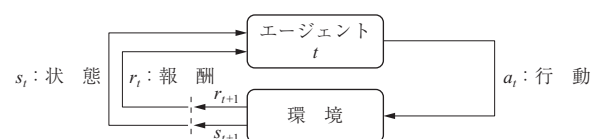
第 2 図は強化学習の基本構造を示したものであり, 環境と, 環境に応じて行動を決めるエージェントで構成される.

エージェントは, t 回目の試行における状態 s_t から次にとる行動 a_t を選択し, 環境へ送る. 環境は, 受け取った行動 a_t によって遷移した状態 s_{t+1} と, 事前に設定したその行動に対する報酬 r_{t+1} をエージェントに送る. 状態に応じた行動の選び方によってさまざまな RL の手法が提案されており, そのなかの一つである Q 学習は, 以下の式で表す Q 関数 (行動価値関数) でその行動の価値を決定し, エージェントは価値が高い行動を選択する.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \{ r_t + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t) \} \quad \dots\dots (5)$$

ここで, α ($0 < \alpha \leq 1$) は学習率, γ ($0 < \gamma \leq 1$) は割引率であり, A は取り得る行動全体を表す. (5) 式は第 2 項が行動価値の期待値と現在の見込み値の差を表しており, この差分だけ現在の行動価値を更新する. 実際に Q 学習を行う際は, Q 関数をテーブル関数で作成し, 取り得る全ての状態と行動の組合わせに対応した価値を保持する必要がある. しかしながら, AlphaGo や自動運転のように状態が画像として得られる場合は, Q 関数のデータ量が膨大になり, 学習を行うことが困難になる. そこで, NN を Q 関数の近似関数として使用したのが DQN である.

DQN では教師あり学習で Q 関数を近似するが, Q 関数の真値は知り得ないので答えを得ることができない. そこで (5) 式の行動価値の期待値を答えとして使用する. この場合, NN の誤差関数 E_{DQN} は以下ようになる.



第 2 図 RL の概念図⁽¹⁸⁾
Fig. 2 Schematic image of RL⁽¹⁸⁾

$$E_{DQN} = \frac{1}{2} \left\{ r_t + \gamma \max_{a \in A} Q_{\theta}(s_{t+1}, a) - Q_{\theta}(s_t, a_t) \right\}^2 \quad \dots\dots\dots (6)$$

ここで、 Q_{θ} は NN で近似した Q 関数を表す。Q 関数の更新量に当たる (5) 式の右辺第二項がゼロになるように学習することで、局所最適な Q 関数が得られる。

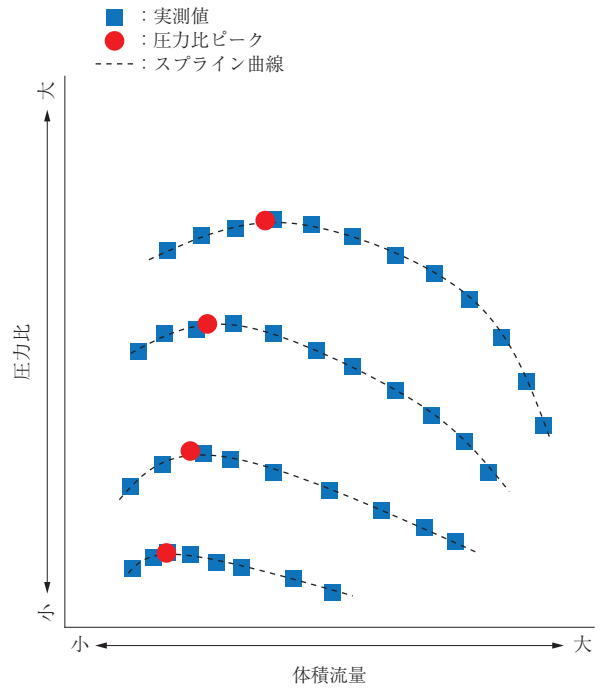
前述のように状態が画像として得られる場合は、**第3図**に示す CNN (Convolutional Neural Network: 畳込みニューラルネットワーク)⁽¹⁹⁾ が用いられる。CNN は特徴量抽出を行う畳込み層と次元削減を行うプーリング層で構成される。畳込み層では、画像処理で用いられるフィルタリングと同様の操作がネットワークの重みを用いて行われ、学習によってフィルタを更新することで重要な特徴量が抽出されるようになる。プーリング層ではある範囲内の平均値や最大値を抽出する処理が行われるが、学習によって更新されることはない。CNN による画像認識では畳込み層やプーリング層の組を複数使用した後で FCNN を用いて出力を得る構造が多い。DQN に CNN を用いる場合、画像から得られた行動価値で次の行動が決まる。これは人が視覚情報を基に自らの行動を学習・選択する過程を模倣した仕組みになっている。

2.2 DNN によるサージ流量予測

本研究では DNN を過給機のサージ流量予測に適用し、実際の製品の過去の試験データを用いて予測モデルを構築した。

2.2.1 使用データ

過去に測定された**第4図**に示すような圧力比-流量特性のデータを用いた。各測定点を近似した3次のスプライン曲線の圧力比ピークにおける体積流量をサージ流量として、サージ流量とその時のサージ点圧力比を取得し、(7)式と(8)式を用いて計算されるサージ流量係数 ϕ_{surge} と圧力係数 μ_{surge} のそれぞれを予測する回帰モデルを



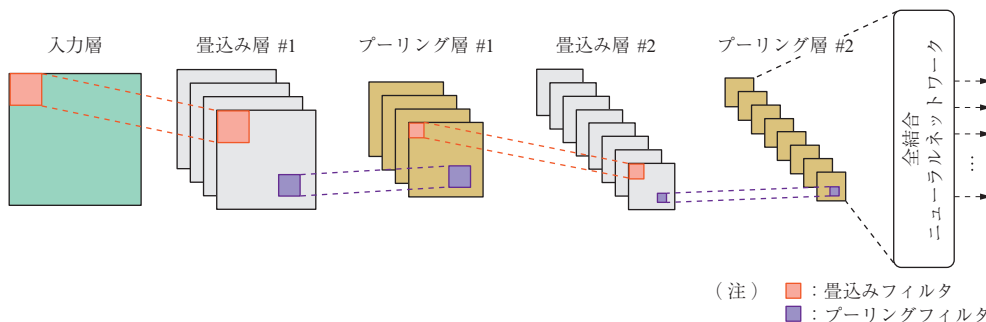
第4図 圧力比-流量特性と圧力比ピーク
Fig. 4 Characteristics between pressure ratio and volumetric flow rate and peak value of pressure ratio

DNN で作成した。

$$Q_{surge} = \phi_{surge} N_{rot} \pi D^3 \quad \dots\dots\dots (7)$$

$$\Pi_{surge} = \left\{ \mu_{surge} \left(Mu \sqrt{T_0 \gamma R} \right)^2 \left(\frac{\gamma - 1}{T_0 \gamma R} + 1 \right) \right\}^{\frac{\gamma}{\gamma - 1}} \quad \dots\dots\dots (8)$$

ここで、 Q_{surge} と Π_{surge} はサージ流量とサージ点圧力比、 N_{rot} は回転数、 D はインペラ外径、 Mu は周速マッハ数であり、 $T_0 = 293.15$ K、 $\gamma = 1.4$ 、 $R = 287$ J/kg·K を用いた。データは 102 機種分あり、機種ごとに 4 ~ 7 条件の周速マッハ数に対応したサージ流量とサージ点圧力比が得られるため、サンプル数は全部で 574 個ある。学習に用いる入力パラメータとして、過給機の知見が豊富な専門家が選定したサージ発生に関係すると考えられる形状パラメータ 100 個、周速マッハ数 1 個、三つの簡易な一次元物理モ



第3図 CNN の構造
Fig. 3 Architecture of CNN

デルによるサージ点流量係数の予測値 3 個の計 104 個を用いた。

DNN に限らず機械学習では、学習結果を評価するためのデータ分割手法が用いられる。その一つであるホールドアウトはデータを学習用データと検証用データに分け、学習用データで学習後に、検証用データで予測精度を検証する手法である。検証用データは学習に使用されないため、未知データに対する予測精度を評価することができる。また、別の分割手法であるクロスバリデーションはデータを複数個に分割し、そのなかの 1 組を除いたデータで学習し、除いた 1 組のデータで予測精度の検証をする⁽²⁰⁾。データの組を入れ替えながら分割した数だけこれを繰り返し、得られた予測精度の平均を最終的な予測精度として評価する。クロスバリデーションは用意したデータ全てを学習に使用でき、学習に用いるパラメータの設定に使用されることが多い。本研究ではホールドアウトを用いて、102 機種から特定の 6 機種（サンプル数 31 個）を検証用データとして使用し、残りの 96 機種（サンプル数 543 個）を学習用データとして使用した。

2.2.2 DNN のハイパーパラメータ

DNN の学習にはノード数や正則化率など多くのパラメータが関わり、ハイパーパラメータと呼ばれる。これらは使用者の経験によって決められるほか、ハイパーパラメータの組合わせを網羅的に試すグリッドサーチ、指定した範囲内でハイパーパラメータの値をランダムに変化させるランダムサーチや最適化手法の一つであるベイズ最適化が用いられる⁽²¹⁾。本研究ではベイズ最適化を用いてハイパーパラメータを決定した。ベイズ最適化では DNN をハイパーパラメータを変数としてもつ予測誤差の関数として扱い、ハイパーパラメータに対応する予測誤差を、ガウス過程回帰などの確率的な回帰手法を用いて近似することで、最も予測誤差が小さくなると期待されるハイパーパラメータを得ることができる。

ベイズ最適化によるハイパーパラメータの決定では前述のクロスバリデーションを使用し、分割数を 5 として、サージ流量の予測値の二乗誤差で評価した。DNN の学習には Chainer ver. 3.1.0⁽²²⁾、ベイズ最適化には GPyOpt ver. 1.0.3⁽²³⁾ を用いた。最適化したハイパーパラメータはエポック数、バッチサイズ、中間層ノード数、中間層層数、ドロップアウト率の五つであり、中間層ノードは全ての中間層で同じ数を用いた。それ以外のハイパーパラメータは固定し、活性化関数は leaky ReLU、学習係数の最適

化手法は Adam、正則化率は $\lambda = 0.0005$ 、重みの初期値分布は LeCun Normal（平均 0、標準偏差 $1/\sqrt{n}$ （ n ：各層のノード数））とした。第 1 表にハイパーパラメータの探索範囲とベイズ最適化で得られた最適値を示す。第 1 表はサージ流量の学習に関して最適化した結果であるが、サージ点圧力比の学習でも同じ値を用いるものとした。

2.2.3 結果・考察

本章の結果は予測したサージ流量係数 ϕ_{surge} と圧力係数 μ_{surge} を (7) 式と (8) 式を用いてサージ流量とサージ点圧力比に変換した値で評価を行ったものである。

はじめに DNN とそのほかの機械学習手法である SVM (Support Vector Machine)、RF (Random Forest)、KRR (Kernel Ridge Regression) の三つによる学習結果を比較して、DNN の有効性を確認する。DNN 以外の手法はランダムサーチを用いてハイパーパラメータを決定し、学習には Scikit-learn ver. 0.19.1⁽²⁴⁾ を使用した。第 2 表に検証用データ 6 機種のサージ流量予測に対する平均二乗平方根誤差 (Root Mean Squared Error: RMSE) を示す。RMSE は以下の式で定義される。

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^M (t_i - y_i)^2} \dots\dots\dots (9)$$

ここで、 M はサンプル数、 t_i と y_i はそれぞれサンプル i の答えと予測値である。第 2 表から検証用データ全体を見ると DNN はほかの機械学習手法と比較して高い精度

第 1 表 ベイズ最適化によるハイパーパラメータの探索範囲と最適値

Table 1 Hyper parameters optimized by Bayesian Optimization

ハイパーパラメータ	単 位	探索範囲	最適値
エポック数	回	1 000 ~ 3 000	2 204
バッチサイズ	個	100 ~ 600	543
中間層ノード数	個	50 ~ 520	450
中間層層数	層	3 ~ 10	6
ドロップアウト率	-	0.000 ~ 0.510	0.316

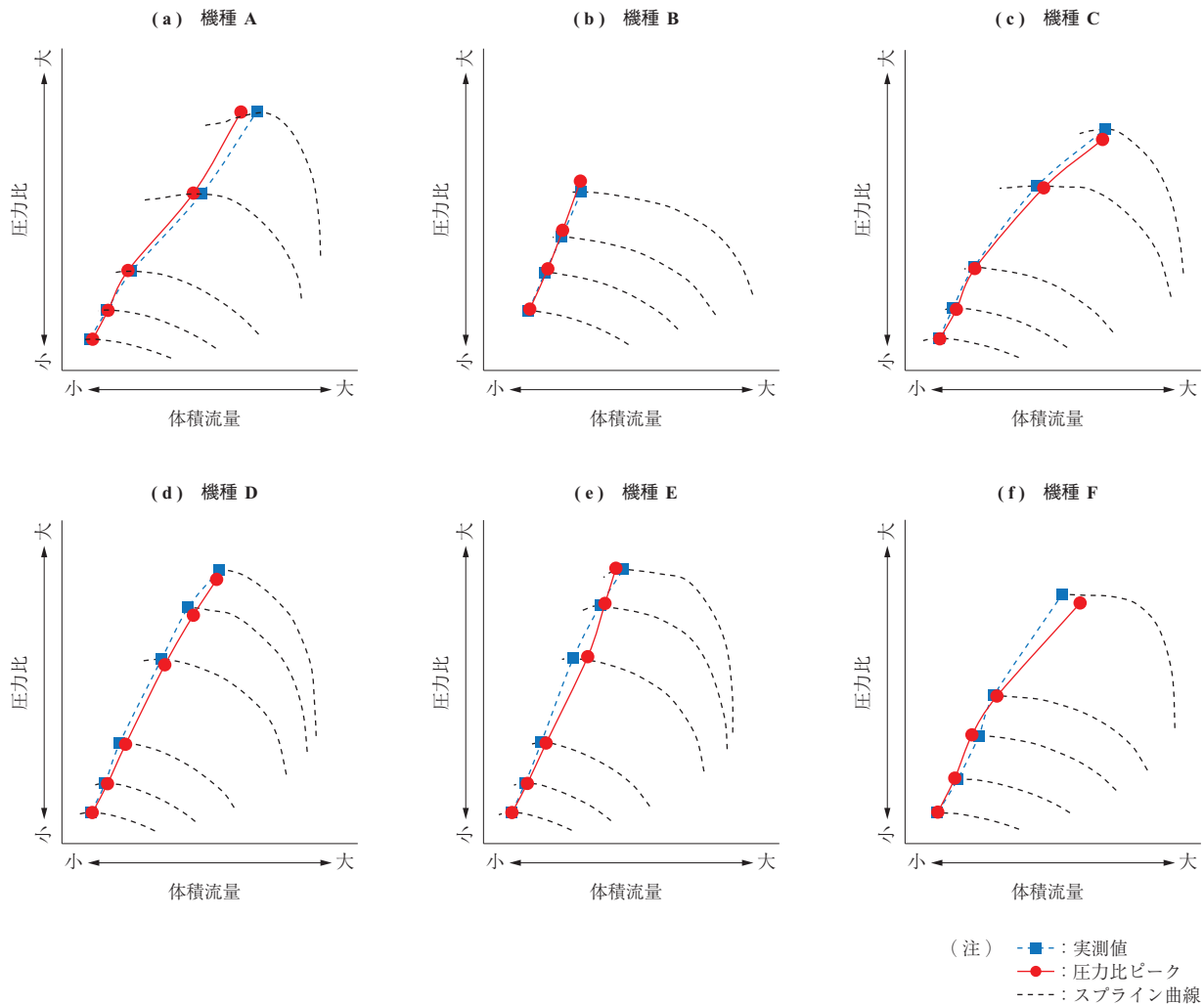
第 2 表 DNN と他の機械学習手法の予測精度比較

Table 2 Performance comparison of DNN and other machine learning methods

機 種	サージ流量の RMSE (kg/m ³)			
	DNN	SVM	RF	KRR
A	0.229 7	0.242 1	0.468 3	0.742 3
B	0.032 4	0.807 7	0.188 5	0.476 8
C	0.083 7	0.276 6	0.316 1	0.663 0
D	0.101 3	0.473 3	0.141 2	0.302 2
E	0.196 8	0.319 5	0.176 5	0.162 9
F	0.387 4	0.442 8	0.352 7	0.447 2
検証用データ全体	0.208 3	0.448 0	0.293 3	0.494 1

で予測できていることが確認できる。機種別に比較しても一部を除いて DNN が最も高精度であり、他の機械学習の方が高精度な場合でも僅差である。以上より、サージ流量予測に関して DNN が有効であることを確認できた。サージ現象は非線形性が強いいため、他手法よりも非線形性を良く表現できる DNN が有効であったと考えられる。

第 5 図に検証用データ 6 機種の圧力比 - 流量特性、サージ流量とサージ点圧力比の実測値と DNN による予測値を示す。予測値は縦軸・横軸ともに、それぞれ DNN で予測した値である。また、圧縮機の運転条件によってサージの発生メカニズムが異なり、予測精度に影響すると考えられる。第 3 表に検証用データ 6 機種のサージ流量、



第 5 図 サージ流量とサージ点圧力比の実測と予測

Fig. 5 Prediction of volumetric flow rate at surge and pressure ratio at surge for test data

第 3 表 サージ流量、サージ点圧力比の RMSE

Table 3 RMSE of prediction of volumetric flow rate and pressure ratio at surge for test data

機 種	サージ流量の RMSE (kg/m ³)			サージ点圧力比の RMSE (-)		
	合 計	Mu < 1.3	Mu > 1.3	合 計	Mu < 1.3	Mu > 1.3
A	0.229 7	0.076 0	0.349 4	0.005 3	0.004 2	0.006 5
B	0.032 4	0.036 5	0.014 7	0.060 9	0.040 3	0.099 8
C	0.083 7	0.048 5	0.118 2	0.046 2	0.012 2	0.071 5
D	0.101 3	0.098 2	0.104 2	0.052 9	0.008 8	0.074 2
E	0.196 8	0.081 9	0.266 0	0.010 5	0.004 6	0.014 2
F	0.387 4	0.180 5	0.787 5	0.036 8	0.006 1	0.081 3
検証用データ全体	0.208 3	0.104 6	0.307 9	0.040 1	0.017 5	0.060 6

サージ点圧力比の機種別の RMSE, それを低速条件 ($Mu < 1.3$) と高速条件 ($Mu > 1.3$) で区別した RMSE を示す. 第 3 表のサージ流量に着目すると, 機種 F が他の機種に比べて予測精度が低いことが分かる. 第 6 図にサージ現象に与える影響が強いと考えられる特定の形状パラメータ α, β のヒストグラムを学習用データから作成したものと, 機種 A ~ F の形状パラメータ α, β がそれぞれヒストグラムのどの領域に属するかを示す. 第 6 図より, 機種 A ~ E と同程度の形状パラメータ α, β の値が学習用データに比較的豊富に含まれているのに対して, 機種 F は少ない, もしくは含まれていないことが確認でき, 学習用データに含まれる相対的なデータ量の違いが予測精度に反映されたと考えられる.

第 3 表のサージ流量, サージ点圧力比の低速条件と高速条件の違いに着目すると, 予測対象や機種によらず高速条件の予測精度が低い傾向にあることが確認できる. 学習用データの低速条件と高速条件のサンプル数の比率は, 低速条件は全サンプルの 62% (339 個) であるのに対し, 高速条件は 38% (204 個) であることから, 高速条件のサンプル数が相対的に少ないことが予測精度の差を生んだ原因の一つだと考えられる. サンプル数に差が出た原因として, 低速条件と比較して高速条件の測定が困難であることが挙げられる.

RMSE は DNN の損失関数と結びついており, 収束の過程を評価するのに適しているが, 予測誤差を直感的に評価するには適していない. そこで, RMSE の代わりに真値に対する相対的な誤差である MRE (Mean Relative Error: 平均相対誤差) で予測性能を評価する. MRE は以下の式で定義される.

$$MRE = \frac{1}{M} \sum_{i=1}^M \left| \frac{t_i - y_i}{t_i} \right| \dots\dots\dots (10)$$

第 4 表に示す検証用データ全体の MRE からサージ流量は約 5%, 圧力比は約 1% の平均相対誤差であり, サージ点圧力比の方が高精度に予測できていることが分かる. これは, 第 7 図に示す今回使用した学習用データにおける周速マッハ数とサージ流量, サージ点圧力比の関係から明らかのように, サージ点圧力比の方が周速マッハ数と相関係数が高いことが原因の一つであると考えられる. 使用しているデータなどが異なるものの, Tamaki⁽²⁵⁾ の一次元物理モデルを用いたサージ流量の予測は MRE で約 10% であるため, 比較的高精度に予測できており, 目標とした MRE 5% をおおむね達成した.

2.3 DQN による LPT 翼の圧力損失最小化

はじめに, 風力発電機や航空機の翼型に用いられる NACA 翼を対象に, 揚抗比を目的関数とした形状最適化の数値実験を行い, DQN の性能検証を行った. その後, LPT (低圧タービン) 翼を対象に圧力損失を目的関数とした形状最適化を行った. DQN の学習には Chainer RL ver. 0.3.0⁽²⁶⁾ を用いた.

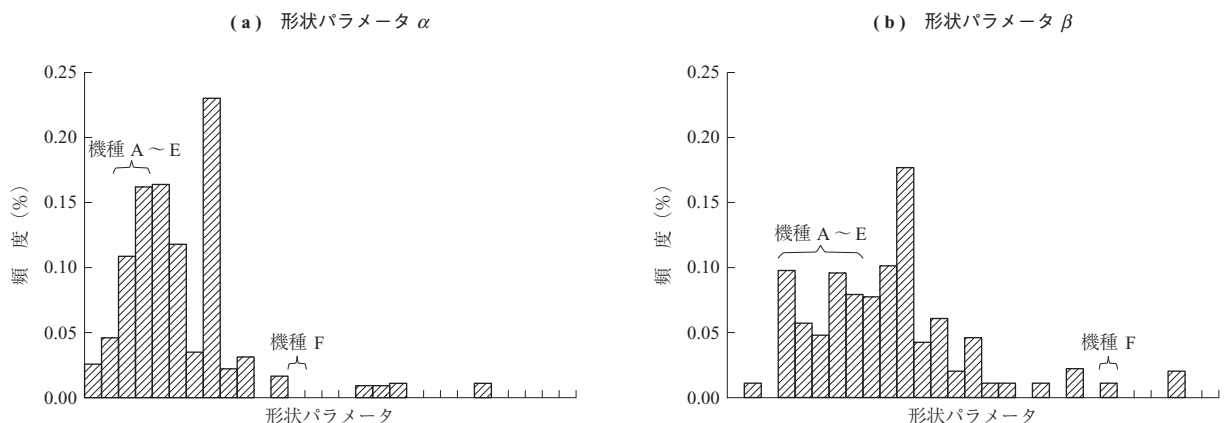
2.3.1 NACA 翼の揚抗比最大化

NACA 翼は, NACA6410 のように名称の数値が形状を一意に決めるパラメータを表す翼型である. NACA 4 桁系列は翼弦長で正規化された最大キャンバ, 最大キャンバ

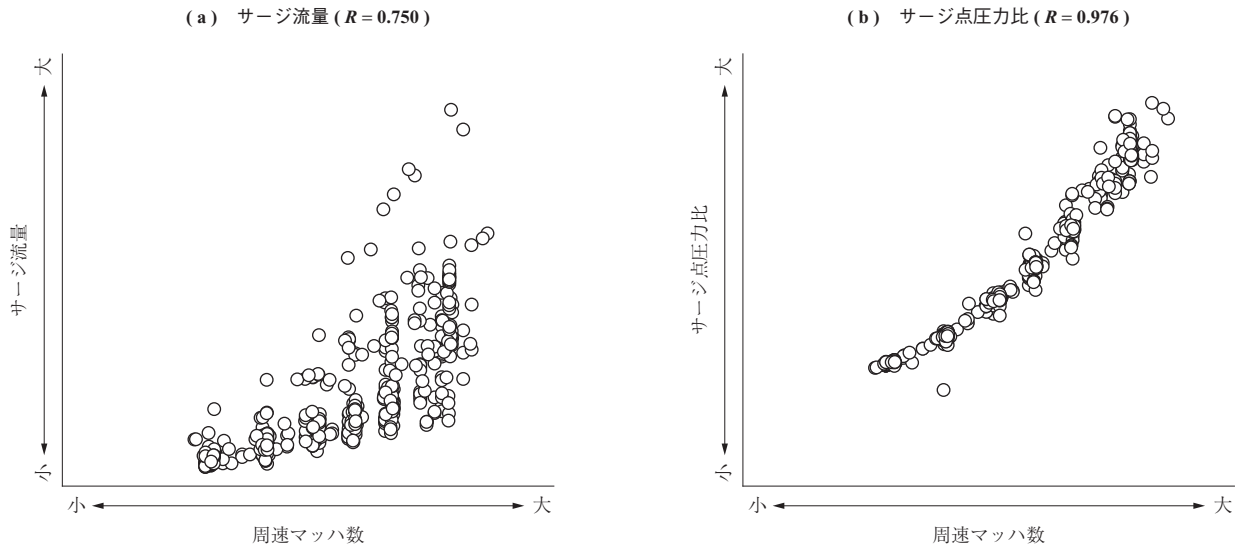
第 4 表 サージ流量とサージ点圧力比の MRE

Table 4 MRE of prediction of volumetric flow rate at surge and pressure ratio at surge for test data

機 種	サージ流量 (%)	サージ点圧力比 (%)
検証用データ全体	5.02	1.38

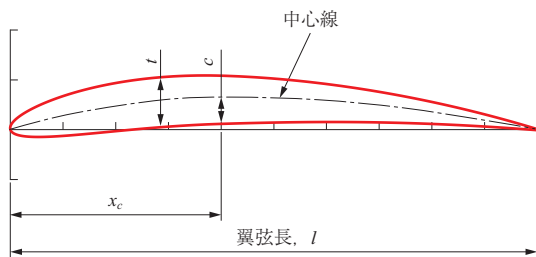


第 6 図 サージに影響する形状パラメータの学習データ内のヒストグラム
Fig. 6 Histograms of shape parameters having an effect on surge



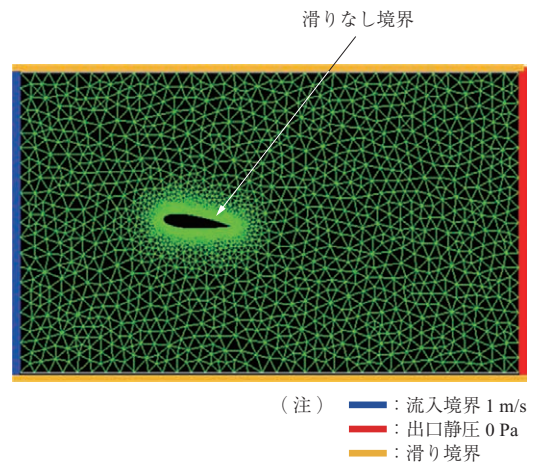
第 7 図 周速マッハ数とサージ流量、サージ点圧力比の相関関係
 Fig. 7 Correlation coefficient R between impeller tip Mach number and prediction targets for training data

位置、最大翼厚の三つの形状パラメータをもち、NACA5桁系列はこれらに中心線の形が加わる。第 8 図に NACA 4 桁系列の一例を示す。ここでは翼型を固定し、望大特性である揚抗比を最大にするように、翼の前縁に流体が当たる角度である迎角の最適化を DQN で行った。エージェントは迎角を増減することの 2 択を選択し、これを行動 a_t と定義する。ただし迎角は 0 度から 40 度の範囲で 1 度単位で変化させる。またエージェントは、外部環境が迎角に応じたメッシュ作成と二次元定常非圧縮 CFD 計算、およびポスト処理を行って出力した 400×400 px の圧力コンター図を状態 s_t 、揚抗比のスカラ値を報酬 r_t として受け取るシステムとした。エージェントは、圧力コンター図を基に迎角の増減を決めるため、2.1.2 項に記した画像認識に特化した CNN を Q 関数の近似に使用した。CFD は OpenFOAM ver. 1706⁽²⁷⁾ を使用し、第 9 図に境界条件を示す。流入境界を流速 1 m/s、出口静圧を



(注) NACA XYZZ
 $X = c/l \times 100$: 最大キャンバ
 $Y = x_c/l \times 10$: 最大キャンバ位置
 $ZZ = t/l \times 100$: 最大翼厚

第 8 図 NACA 4 桁系列
 Fig. 8 Schematic image of NACA 4-digits



第 9 図 境界条件
 Fig. 9 Boundary conditions

0 Pa、翼面を滑りなし境界、解析領域の上下境界を滑り境界とし、乱流には k-ε モデルを用いた。また、翼弦長 1 m、流体密度 1 kg/m^3 とし、 Re (Reynolds 数) が 10^3 、 10^5 、 10^7 になるように流体粘度 1.0×10^{-3} 、 1.0×10^{-5} 、 $1.0 \times 10^{-7} \text{ Pa}\cdot\text{s}$ の 3 条件で DQN による学習を行った。 Re が 10^3 の流れ場は層流となり、k-ε モデルを使用することは妥当ではなく、また、20 度以上の高迎角となる剥離域では発生する渦の大きさと格子解像度に合わせて適切に乱流モデルを選択する必要がある。しかしながら、ここでの目的は DQN の性能検証であるため、システムを簡略化するために揚抗比が最大となる迎角 5 ~ 20 度付近に合わせて格子解像度と乱流モデルを決定した。学習では初期の迎角をランダムに決定し、エージェントが 50 回迎角を変えるまでを 1 セットとし、400 セットの学習を行った。

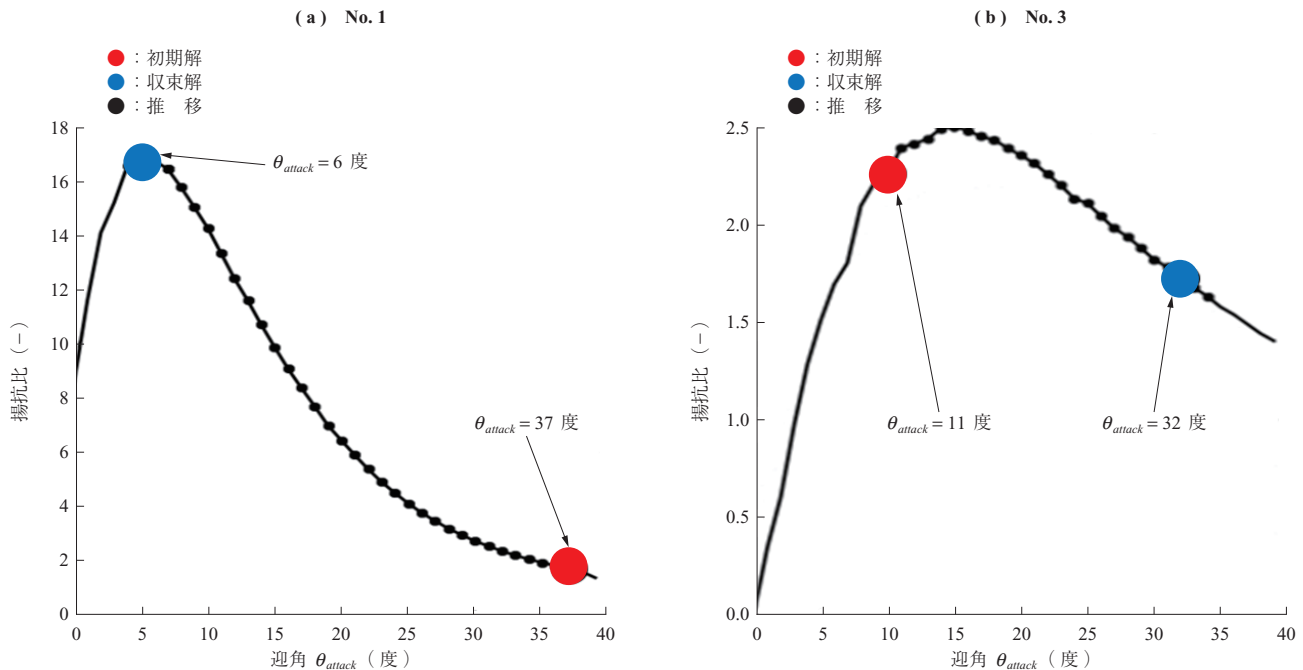
与える報酬として、迎角を変えて揚抗比が上昇すれば $r_t = +1$, 低下すれば $r_t = -1$ を与えた。ここでは数値実験として、第5表に示す四つの学習と検証の組み合わせの実験を行い、DQNによる最適化性能を評価した。第5表の四つの数値実験は、No. 1が学習時と検証時に同じ4桁系列を使用するが、翼形状は異なる場合、No. 2が学習時と検証時に翼の系列が異なる場合、No. 3が学習時と検証時に翼形状と Re が異なる場合、No. 4が低速と高速の条件

第5表 数値実験条件
Table 5 Conditions of numerical experiments

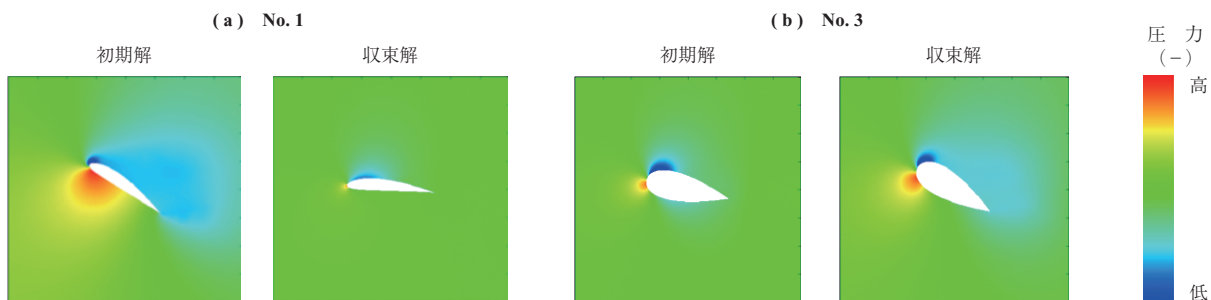
No.	学 習		検 証	
	NACA	Re	NACA	Re
1	4桁系列	10^3	4桁系列	10^3
2	4桁系列	10^3	5桁系列	10^3
3	4桁系列	10^3	4桁系列	10^5
4	4桁系列	$10^3, 10^7$	4桁系列	10^5

で学習し、中速条件で検証した場合である。

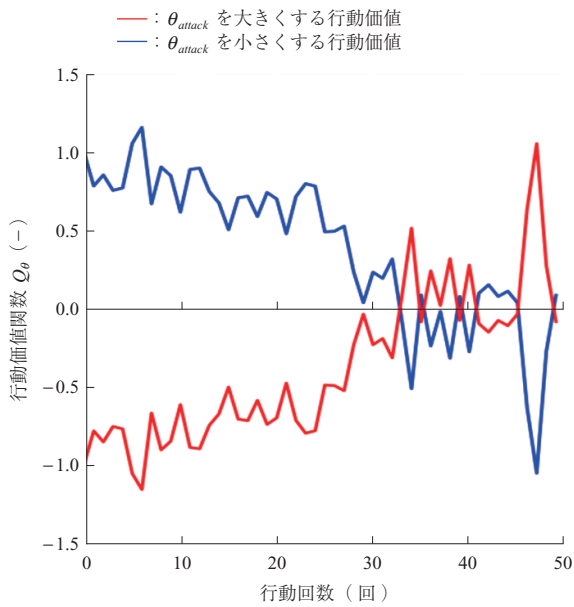
第10図-(a)にNo.1のある翼形状における検証時の初期解から収束解への推移を示す。迎角37度の初期条件に対し、徐々に迎角を下げて迎角6度に収束していることが分かる。第11図-(a)は第10図-(a)の初期解と収束解の圧力コンターである。初期解は翼の前縁部で圧力が低くなり、剥離が生じているが、収束解では圧力が低い領域が減少していることが確認でき、エージェントが圧力コンター上で剥離を認識し迎角を小さくするように行動したと考えられる。第12図にNo.1の学習時のあるセットにおける行動価値関数 Q_θ の推移を、横軸を1セット中の行動回数として示す。行動回数30回程度までは迎角を小さくする行動価値が迎角を大きくする行動価値よりも高い。これは初期の迎角が大きいため迎角を小さくする行動をとれば良いと明確に判断できている状態を示す。行動回



第10図 初期解から収束解への推移
Fig. 10 Transition of lift-to-drag ratio from initial solution to converged solution



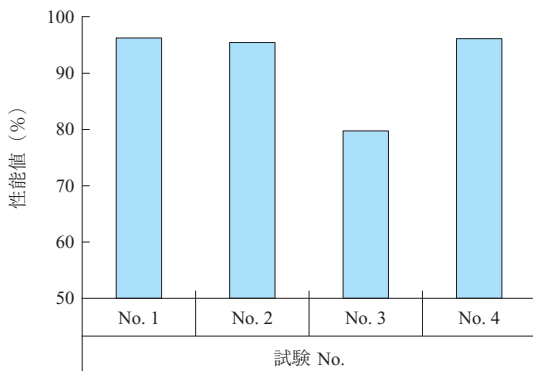
第11図 初期解と収束解の圧力コンター
Fig. 11 Pressure contour map of initial and converged solutions in Fig. 10



第 12 図 行動価値関数の推移 (No. 1)
Fig. 12 Transition of Q function of No. 1

数 30 回以降では両者の行動価値の差が小さくなり、これは収束解近傍に到達したことで、迎角が振動していることを意味している。

No. 1 ~ 4 それぞれの DQN の性能を評価するために、エージェントの行動が収束した迎角における揚抗比を、エージェントの探索範囲内の揚抗比の最適値で割った値を算出した。この値の翼形状 20 個の平均値を第 13 図に示す。縦軸の性能値は大きいほど DQN の性能が高いことを表す。No. 2 は検証時に NACA 5 桁系列を使用しているが、 Re は変化させていないので No. 1 と同程度の性能であることが確認できる。それに対し、検証時に Re を変化させた No. 3 は No. 1 と No. 2 と比較して性能値が 17 ポイント程度低い。これは Re が異なるため検証で得られる圧力コンターに学習時にはなかった傾向が表れたためと考えられ、流れ場が変わると適切に最適化できないこ

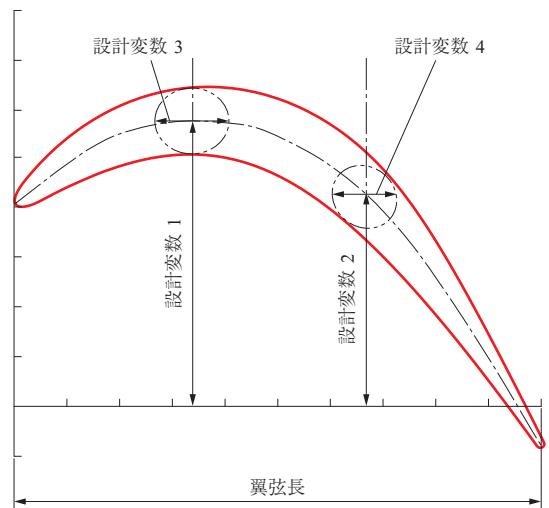


第 13 図 各試験の評価値
Fig. 13 Performance comparison among each numerical experiment

とを示している。第 10 図 - (b) に No. 3 のある翼形状における検証の初期解から収束解への推移を、第 11 図 - (b) に第 10 図 - (b) の初期解と収束解の圧力コンターを示す。この翼形状の最適な迎角は 15 度であるのに対し、第 10 図から 32 度に収束していることが確認できる。また、第 11 図 - (b) から初期解で生じている前縁部の剥離が収束解でも残っていることが分かる。一方で、学習時に使用していない Re であっても、学習条件の内挿域に入る No. 4 は No. 1 や No. 2 と同程度の性能値になることが第 13 図から確認できる。以上から、より幅広い範囲の条件で DQN の学習を行うことで、学習済みのエージェントを流用することができると考えられる。

2.3.2 LPT 翼の圧力損失最小化

ジェットエンジン向け低圧タービンの翼形状である PACK B⁽²⁸⁾ を基準とした形状最適化を行い、圧力損失の最小化を行った。設計変数は翼弦長の 1/3 と 2/3 の位置のキャンバ長と翼厚の四つとし、第 14 図に示す。ここでは流れ場の条件を固定し、望小特性である圧力損失を制約条件のもと最小にするように、四つの設計変数の最適化を DQN で行った。エージェントは、外部環境が設計変数に応じたメッシュ作成と、二次元定常非圧縮 CFD 計算およびポスト処理を行って出力した 400×400 px のマッハ数コンター図を、状態 s_t として受け取るシステムとした。エージェントの Q 関数の近似には CNN を使用し、マッハ数コンター図を基に四つの設計変数を変化させる行動



(注) 設計変数 1: 翼弦長 1/3 位置のキャンバ長
設計変数 2: 翼弦長 2/3 位置のキャンバ長
設計変数 3: 翼弦長 1/3 位置の翼厚
設計変数 4: 翼弦長 2/3 位置の翼厚

第 14 図 LPT 翼の設計変数
Fig. 14 Design variables of LPT airfoil

a_t をとる. DQN に制約条件を課す場合, 報酬を定義する際に目的関数と制約条件を合わせて考慮する必要がある. ここでの目的関数は圧力損失であり, 制約範囲は出口角 -58.3 ± 0.2 度とした. そこで, 報酬として, (11) 式に示す評価関数 f を定義し, 評価関数が増加した時に $r_t = +1$, 低下した時に $r_t = -1$ の報酬を与えることとした.

$$f = -cP_{\text{drop}} - |\theta_{\text{const}} - \theta| \quad \dots\dots\dots (11)$$

ここで, P_{drop} は圧力損失 [-], $\theta_{\text{const}} = -58.3$ 度は制約条件の出口角, θ は CFD によって得られる出口角である. (11) 式の第 1 項が目的関数, 第 2 項が制約条件であり, 定数 $c = 500$ によって重み付けを行っている. 圧力損失 P_{drop} は以下の式で計算する.

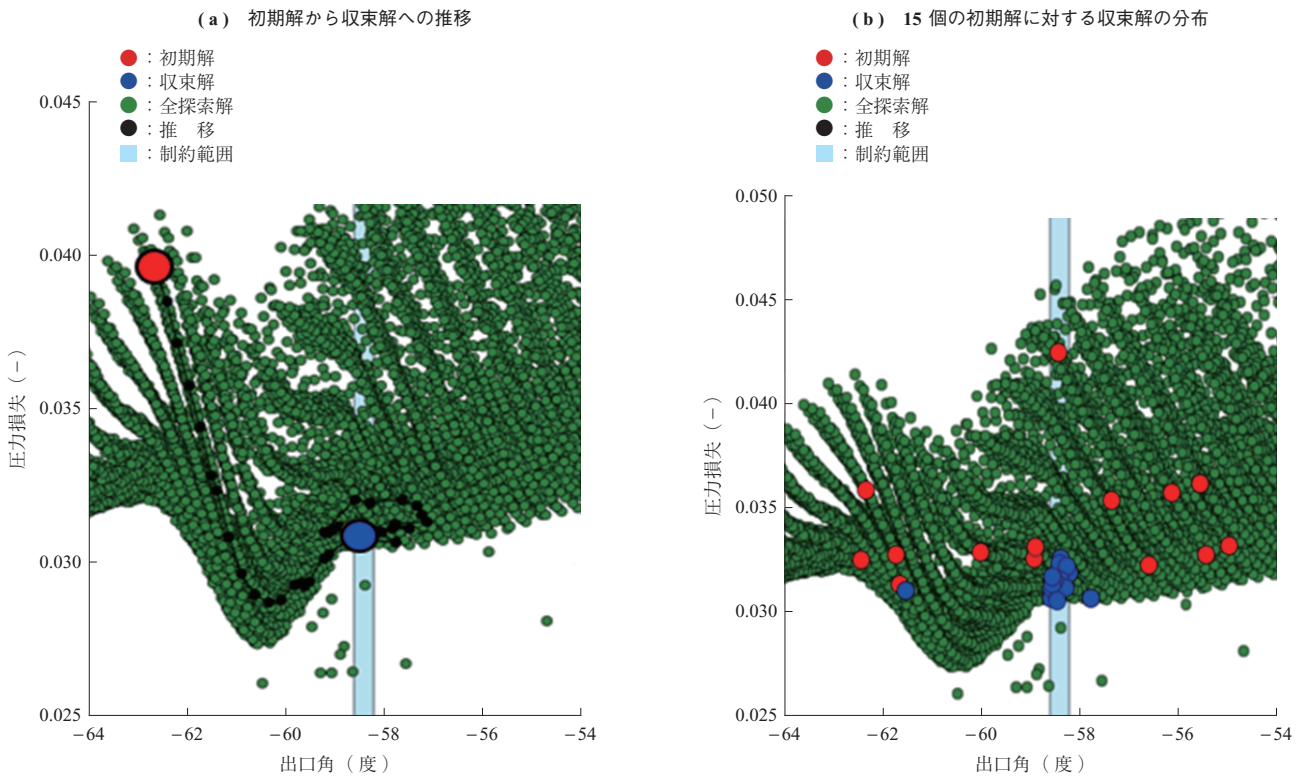
$$P_{\text{drop}} = \frac{P_{\text{Inlet}}^T - P_{\text{Outlet}}^T}{P_{\text{Outlet}}^T - P_{\text{Outlet}}^S} \quad \dots\dots\dots (12)$$

ここで, P_{Inlet}^T は入口全圧, P_{Outlet}^T は出口全圧, P_{Outlet}^S は出口静圧であり, それぞれ CFD で得ることができる. 学習では初期形状をランダムに決定し, 50 回形状を変えるまでを 1 セットとし, 200 セットの学習を行った.

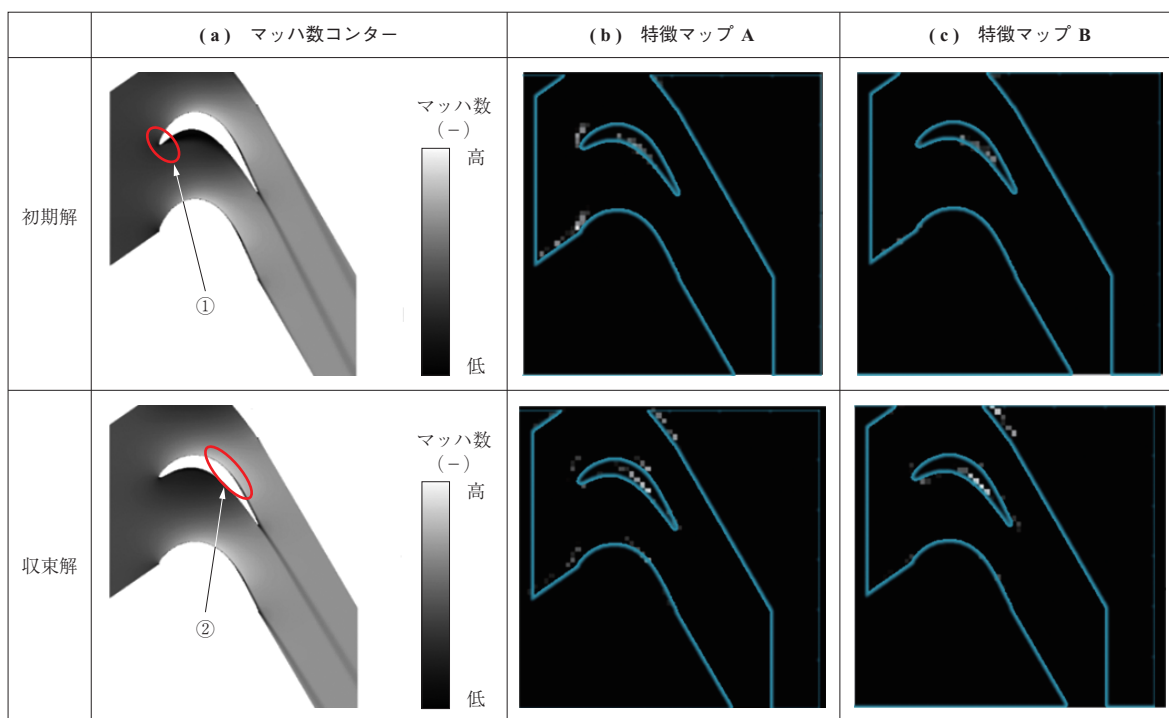
第 15 図に学習後のエージェントを用いて, ある初期解の最適化を行った際の収束解への推移 (- (a)) と 15 個の初期解に対する収束解の分布 (- (b)) を示す. 横軸は制約条件の出口角, 縦軸は目的関数の圧力損失であり, 図

中に示された制約条件の領域内で最小の圧力損失となるものが最適解である. 第 15 図には約 12 000 個の形状候補に対応する点も併せてプロットした. 第 15 図 - (a) から制約条件の範囲内で解が収束していることが確認できる. また, 第 15 図 - (b) からランダムに生成された初期形状に対し, ほとんどが制約条件の範囲内で最適解に収束していることが分かる.

第 16 図 - (a) は第 15 図 - (a) の初期解と収束解のマッハ数コンターである. 一般的に翼設計者がマッハ数コンターを基に翼設計を行う場合, 第 16 図 - (a) に示す領域 ①と②に着目し, 圧力損失は背側後縁部の剥離, 出口角は前縁部の速度分布から評価する. そこで, エージェントが状態として受け取ったマッハ数コンターのどこに着目して行動を決定しているかを理解するため, Q 関数を近似した CNN に入力されたコンター図の特徴マップ (畳み込み層でフィルタリングされた後の画像) を確認する. 第 16 図 - (a) のマッハ数コンターに対する特徴マップは CNN のなかから複数枚取得できるが, そのなかから二つ選択した特徴マップ A と B を第 16 図 - (b), - (c) に示す. 特徴マップを見やすくするために, 便宜的に翼形状と計算領域を示した. 特徴マップで明るく表示されている箇所がフィルタリングによって抽出されたマッハ数コン



第 15 図 LPT 翼圧力損失最小化の初期解と収束解
Fig. 15 Initial solutions and its converged solutions



第 16 図 第 15 図 - (a) の初期解と収束解のマッハ数コンターと特徴マップ
 Fig. 16 Mach number contour and feature maps of initial and converged solutions of Fig. 15 - (a)

ターの特徴的な部分である。第 16 図 - (b), - (c) のそれぞれの特徴マップから、初期解では前縁部と背側後縁部に特徴が表れていたのに対し、収束解ではこの特徴が消えていることが分かる。これらのことから、それぞれの領域に表れた特徴を消すようにエージェントが行動していると考えられる。特徴が変化した領域は第 16 図 - (a) に示した設計者の着目点である領域 ①と②に対応しており、エージェントと設計者の着目点は一致しているといえる。

3. 結 言

物理現象が未解明な部分を補う、機械学習による性能予測手法と強化学習による設計解探索手法を構築するために、DNN による車両用過給機の圧縮機のサージ流量予測と、DQN による LPT 翼の圧力損失最小化を行った。

DNN による圧縮機のサージ流量予測に関して、学習には過去に測定済みのデータを利用したため、低コストで予測モデルを構築でき、未知データに対し約 5%の相対誤差で予測することができた。予測に用いる入力パラメータの値が学習用データに含まれず、外挿になる条件では予測精度が低下するため、学習用データを拡充し、外層領域を可能な限り減らすことが重要である。本手法で構築した予測モデルは、高速かつ低コストで予測可能であり、設計プロセスの上流で活用が見込まれる。

DQN による最適化に関して、数値実験により学習した条件内であればエージェントは適切に学習されることを確認した。また、学習した条件の範囲外であっても、その条件が内挿域にあれば、DQN による最適化が可能であることが明らかになった。このことは、十分な設計領域をカバーするように学習が行われていれば、多種多様な問題に合わせた最適化が可能であることを示唆している。LPT 翼の圧力損失最小化では、制約条件がある問題に対し、報酬を適切に設定することで最適化が可能であることを確認した。この手法では任意の報酬を定義できるため、ほかの問題に応用することが可能である。また、Q 関数を近似した CNN の畳み層の特徴マップから、エージェントと翼設計者の着目点が一致することが確認できた。本検討のように、機械学習が学習した設計のポイントを設計者にフィードバックすることで、機械学習モデルの妥当性を確認するとともに、設計者に新たな気付きを与えられる可能性がある。

参 考 文 献

- (1) K. Deb : Multi-objective Optimization using Evolutionary Algorithm, John Wiley & Sons, (2005)
- (2) R. H. Myers and D. C. Montgomery : Response Surface Methodology: process and product

- optimization using designed experiments, second edition, A Wiley-Interscience publication, (2002)
- (3) S. Urolagin, K. V. Prema and N. V. Subba Reddy : Generalization Capability of Artificial Neural Network Incorporated with Pruning Method, International Conference on Advanced Computing, Networking and Security, (2011), pp. 171 – 178
- (4) G. E. Hinton, S. Osindero and Y. W. Teh : A Fast Learning Algorithm for Deep Belief Nets, Neural Computation, Vol. 18, (2006), pp. 1527 – 1554.
- (5) V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller : Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop 2013
- (6) U. Kesgin : Genetic Algorithm and Artificial Neural Network for Engine Optimisation of Efficiency and NO_x emission, Fuel, Vol. 83, Iss. 7 – 8, (2003. 10), pp. 885 – 895
- (7) O. Ozener, L. Yuksek and M. Ozkan : Artificial Neural Network Approach to Predicting Engine-out Emissions and Performance Parameters of a Turbo Charged Diesel Engine, Thermal Science, Vol. 17, No. 1, (2013. 1), pp. 153 – 166
- (8) B. Liu, C. Zhao, F. Zhang, T. Cui and J. Su : Misfire Detection of a Turbocharged Diesel Engine by using Artificial Neural Networks, Applied Thermal Engineering, Vol. 55, No. 1, (2013. 6), pp. 26 – 32
- (9) E. Mese and D. A. Torrey : An Approach for Sensorless Position Estimation for Switched Reluctance Motors using Artificial Neural Networks, IEEE Transactions on Power Electronics, Vol. 7, Iss. 1, (2002), pp. 66 – 75
- (10) U. Atici : Prediction of the Strength of Mineral Admixture Concrete using Multivariable Regression Analysis and an Artificial Neural Network, Expert Systems with Applications, Vol. 38, Iss. 8, (2011. 8), pp. 9609 – 9618
- (11) X. Glorot, A. Bordes and Y. Bengio : Deep Sparse Rectifier Neural Networks, In Proc. 14th International Conference on Artificial Intelligence and Statistics 2011, pp. 315 – 323
- (12) A. Maas, A. Hannun and A. Ng : Rectifier nonlinearities improve neural network acoustic models, ICML 2013
- (13) D. P. Kingma and J. Ba : Adam: A Method for Stochastic Optimization, Proceedings of the 3rd ICLR 2015
- (14) N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov : Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research, Vol. 15, (2014), pp. 1929 – 1958
- (15) D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis : Mastering the Game of Go with Deep Neural Networks and Tree Search, Nature, Vol. 529, (2016. 1), pp. 484 – 489
- (16) M. Vecerik, T. Hester, J. Schöolz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothorl, T. Lampe and M. Riedmiller : Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards, arXiv:1707.08817v2, (2018)
- (17) A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V. D. Lam, A. Bewley and A. Shah : Learning to Drive in a Day, arXiv:1807.00412v2, (2018)
- (18) R. S. Sutton and A. G. Barto : Reinforcement Learning: An Introduction, The MIT Press, (2017. 11)
- (19) A. Krizhevsky, I. Sutskever and G. E. Hinton : ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012, pp. 1097 – 1105
- (20) C. Bishop : Pattern Recognition and Machine Learning, Springer, (2006)
- (21) J. Snoek, H. Larochelle and R. Adams : Practical Bayesian Optimization of Machine Learning Algorithms, NIPS 2012, pp. 2951 – 2959
- (22) Chainer : A flexible framework for neural networks, <https://chainer.org>, (参照 2018-10-30)
- (23) GPyOpt : <https://sheffieldml.github.io/GPyOpt/index.html>, (参照 2018-10-30)
- (24) Scikit-learn : machine learning in python, <http://>

scikit-learn.org/stable/, (参照 2018-10-30)

(25) H. Tamaki : Effect of Piping Systems on Surge in Centrifugal Compressors, Journal of Mechanical Science and Technology, Vol. 22, Iss. 10, (2008. 10), pp. 1 857 - 1 863

(26) Chainer RL : <https://github.com/chainer/chainerrl>, (参照 2018-10-30)

(27) OpenFOAM : <http://www.openfoam.org>, (参照 2018-10-30)

(28) J. P. Lake : Flow Separation Prevention on a Turbine Blade in Cascade at Low Reynolds Number, PhD Thesis, Air Force Institute of Technology, (1999. 6)