深層強化学習を用いた翼形状の最適化と手法の比較

The optimization and comparison of methods for the Air foil design using Deep Reinforcement Learning.

服部 均[*1]
Hitoshi Hattori

米倉 一男[*1]
Kazuo Yonekura

[*1] 株式会社 IHI
IHI Corporation

When designing turbomachinery such as jet engines and superchargers, CAE is indispensable technology. In order to generate high performance shapes, the optimization methods such as response surface methodology and genetic algorithm has been used for design . However, these methods require many iterative calculations. When searching for a high performance shape against multiple flow conditions, it is necessary to repeat analysis every time the flow conditions are changed, which lengthens the design time. In this paper, to shorten this design time, we propose a new design method using deep reinforcement learning and compare of methods.

1. 諸言

航空エンジンや車両用過給機をはじめとする機械製品を設計する場合に，CAE（Computer Aided Engineering）は欠くことのできない技術である．設計者は CAE を用いて，例えば翼の周りの流体の流れを数値的に解析し，その物理現象を理解して実際に製品を作製した時の性能等を予測する．所望の制約条件を満たしたうえで最も性能が良い形状を作製するため，これまでは遺伝的アルゴリズム[Kalyanmoy 2005]や応答曲面法[Raymond 2001]などを用いた最適化が行われてきた．これらの手法は多くの繰り返し計算が必要である．流れの境界条件や設計変数の増加などの仕様の変更があった場合には計算をやり直す必要があり，設計に多くの時間を要していた．本稿では，この設計時間を短縮するために，深層強化学習を用いた形状最適化を提案し，手法の比較を行うことで設計に有用な手法を選定する．設計作業では，仕様変更に合わせて，流れの条件を少し変えて形状を検討する作業が繰り返し行われる場合がある．このような場合，多数ある条件を包絡する条件であらかじめ多数の数値計算を実行して学習しておき，実際の設計時は学習済みモデルを用いて検討を行なうことで，実際の設計時間を短縮できると期待される．

2. Deep Q-Network

強化学習（RL; Reinforcement Learning）では，特定の環境を与え，得られる報酬が最大になるように行動を学習する．強化学習については[Richard 1998]が詳しい．強化学習は図 1 に示すように，報酬を与える環境と，エージェントから構成される．エージェントが行動を起こした結果，環境側から報酬と現在の状態が出力される．このエージェントが現在の状態に応じて，最も多くの報酬が期待される行動をとるように学習が進む．

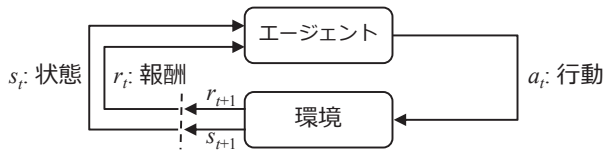

図 1. RL の概念図[Richard 1998]

連絡先：服部均，株式会社 IHI，hitoshi_hattori@ihi.co.jp

強化学習は一般にロボット等の制御やテレビゲームに対して使用されており，時間変化する動的な問題に対して使用されることが多い．一方で文献[Li 2017], [Andrychowicz 2016]ではニューラルネットワークのハイパーパラメータの最適化などの時間変化しない静的な問題における最適化に適用している．本報告で扱う問題も時間変化のない静的な問題である．行動選択の方法について様々な手法が提案されているが，本報告ではその 1 つである Deep Q-network (DQN) [Volodymyr 2013]を用いる．DQN は，次式で表す行動価値関数（Q 関数）を元に行動を決定する．さらにこの Q 関数を深層ニューラルネットワーク(DNN)でモデル化する．

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

ここで，α（0<α≤1）は学習率，γ（0<γ≤1）は割引率であり，A は取り得る行動全体を表す．式（1）は第 2 項が行動価値の期待値と現在の見込みの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．Q 関数を DNN でモデル化する利点の一つが，状態量として画像を扱えることである．

3. NACA 翼の揚抗比最大化

航空機の翼型に用いられる NACA 翼を対象に揚抗比の最大化を目的とした数値実験を行い，手法ごとに性能の比較を行った．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．図 2 に NACA4 桁系列の一例を示す．NACA4 桁系列は翼弦長で正規化された最大キャンバ，最大キャンバ位置，最大翼厚の 3 つの形状パラメータを持つ．ここでは翼型を固定し，膨大特性である揚抗比を最大にするように，迎角の最適化を行った．エージェントが迎角を増減することを図 1 に示す行動a_tとした．迎角は 0 度から 40 度の範囲で 1 度単位で変化させる．また環境は翼形状と迎角を入力として，2 次元定常非圧縮流れの CFD（Computational Fluid Dynamics）計算を行い，圧力コンター図と揚効比を出力するシステムとした．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，Q 関数のモデル化には CNN [Alex 2012]を使用した．深層強化学習の実装には ChainerRL [ChainerRL 2018]を用い，CFD の計算には OpenFOAM [OpenFOAM 2018]を使用した．境界条件は図 3 に示すように，

流入境界(青)を流速 1 m/s, 出口境界(赤)を静圧 0 Pa, 翼面(水色)を滑り無し境界, 解析領域の上下境界(黄)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, 流体粘度 1.0×10-5 Pa·s とした. 学習時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報酬 r_t=+1, 低下すれば r_t= -1 を与えた.

図 4 にある翼型における検証時の初期解から収束解への推移を示す. θ_{attack}は迎角であり, 図 4 は迎角 37deg の初期条件に対し, 徐々に迎角を下げて迎角 6deg に収束していることを示している. 図 5 は初期解と収束解の圧力コンターである. 初期解で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すると迎角を小さくするように行動したと考えられる.

次に強化学習の手法を変えて試行回数と得られる揚抗比の最適値との比の比較を行った. 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評価値として使用した. 比較対象は DQN, DoubleDQN [Hado 2016], ResidualDQN, [Chainer RL 2018], Dynamic Policy Programming (DPP) [Mohamma 2012] , Advantage Learning (AL) [Marc 2016], Persistent Advantage Learning (PAL) [Marc 2016], DoublePAL [Marc 2016], SARSA [Rummery 1994]である. 学習に用いた CFD 計算の回数と各手法の評価値を図 6 に示す. 少ない CFD の回数で安定して高い評価値が得られる手法が望ましい. CFD 計算の回数が 2,500 回以下の場合には AL が最も高い性能を示しているが, 5,000 回付近で評価値が1ポイント低下し, 学習回数によって得られる解が変わりうることがわかる. DQN は 2,500 回以降でも高い評価値を維持していることから本例題においては最も有用な手法であるといえる. 一般的にレトロゲームで DQN より高性能とされる DoubleDQN は CFD の回数が 15,000 回以下の場合には 95%以下の評価値となっていることから, DQN と同程度の精度を得るためには 20,000 回程度の CFD 計算が必要であり, 本課題には適さないことがわかった. その理由として, 本例題は1変数の最適化であり, 問題が単純であったため, シンプルな学習手法が適していた可能性が挙げられる. したがって目的関数が多峰性である場合や設計変数が多い場合は DQN 以外の手法が有用である可能性がある. 本比較により強化学習を用いて翼形状の最適化を行う上で, 選択する手法により必要な CFD の回数に大きな差があり, 本例題においては DQN が最も適していることを確認した. この結果は今後強化学習を用いて最適化設計を行う上で, 手法選択に役立てることができる.

4. 結論

深層強化学習を NACA 翼の揚抗比最大化に適用し, 目的に応じてエージェントが適切に迎角を変更できることを確認した. また手法ごとに学習に必要な CFD の回数を比較し, 本課題では DQN が最も適していた. 本比較により選択する手法次第で必要な CFD の回数に大きな差があることがわかり, 今後強化学習を用いて最適化設計を行う上で, 選択するべき手法の指針が示された.

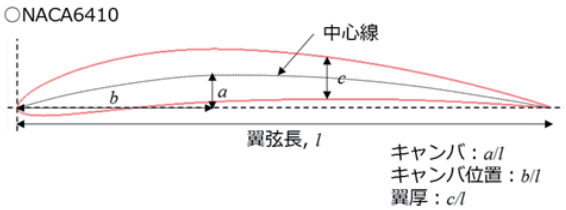

図 2. NACA4 桁系列の翼形状の例

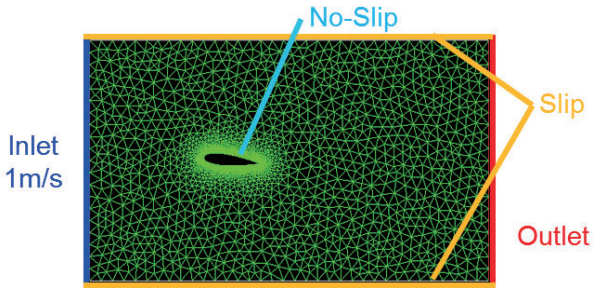

図 3. 境界条件

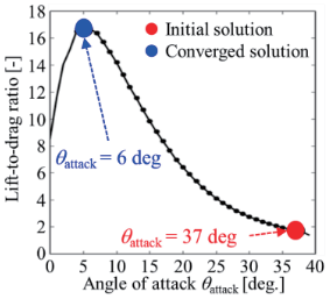

図 4. 初期解から収束解への推移

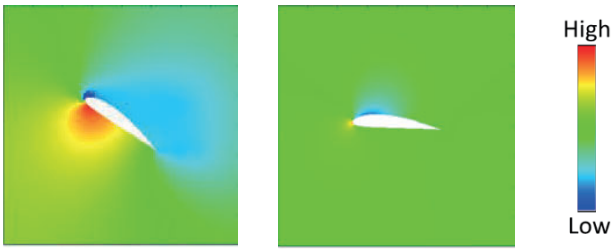

図 5. 初期解と収束解の圧力コンター

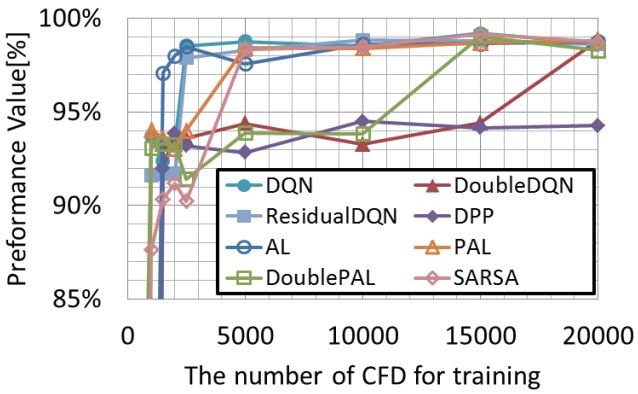

図 6. 各手法の評価値

参考文献

[Kalyanmoy 2005] Kalyanmoy, D.: Multi-objective Optimization using Evolutionary Algorithm, John Wiley & Sons, 2005.

[Raymond 2001] Raymond, H. M.: Response Surface Methodology: process and product optimization using designed experiments, second edition, A Wiley-Interscience publication, 2001.

[Li 2017] Li, K., Malik, J. : Learning to optimize. In: ICLR 2017 conference, 2017.

[Andrychowicz 2016] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., de Freitas, N.: Learning to learn by gradient descent by gradient descent, arXiv preprint arXiv, 1606.04474v2, 2016.

[Richard 1998]Richard S. S. and Andrew G. B.: Reinforcement Learning: An Introduction, MIT Press, 1998.

[Volodymyr 2013]Volodymyr M., Koray K., David S., Alex G., Ioannis A., Daan W., Martin R. :Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop, 2013.

[Alex 2012]Alex K., Ilya S. and Geoffrey E. H.: Imagenet Classification with Deep Convolutional Neural Networks, NIPS, pp.1097–1105, 2012.

[OpenFOAM 2018] OpenFOAM, http://www.openfoam.org, 2018/10/30.

[Chainer RL 2018] Chainer RL, https://github.com/chainer/chainerrl, 2018/10/30.

[Marc 2016] Marc G. B., Georg O., Arthur G., Philip S. T., Remi M.: Increasing the Action Gap: New Operators for Reinforcement Learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[Hado 2016] Hado V. H., Arthur G., David S.: Deep Reinforcement Learning with Double Q-learning, Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[Mohamma 2012] Mohammad G. A., Vicenc G., Hilbert J. K.: Dynamic Policy Programming, Journal of Machine Learning Research 13, 2012.

[Rummery 1994] Rummery, G. A., and Niranjan, M.: On-line Qlearning using Connectionist Systems, Technical Report, Cambridge University Engineering Department, 1994.