〔特集〕注目研究 in CFD31

円柱カルマン渦列の制御における深層強化学習の試行

宇宙航空研究開発機構（現　みずほ情報総研株式会社）　小　泉　　拓
宇宙航空研究開発機構　堤　　誠　司*
宇宙航空研究開発機構　嶋　　英　志

Application of Deep Reinforcement Learning for Feedback Control of a Circular Cylinder Wake

Hiroshi Koizumi, Japan Aerospace Exploration Agency (Currently, Mizuho Information & Research Institute, Inc.)
*Seiji Tsutsumi, Japan Aerospace Exploration Agency
Eiji Shima, Japan Aerospace Exploration Agency
*E-mail for correspondence: tsutsumi.seiji@jaxa.jp

1　はじめに

近年，深層学習は音声認識[1]，画像識別[2]といった分野でそれまでの性能限界を大幅に凌駕し，動画から猫や人間の顔を教師信号無しに識別できるようになる[3]など，画期的な成果を上げている．また，強化学習と深層学習を組み合わせた深層強化学習である Deep Q-Network (DQN)を利用し，単純なテレビゲームにおいて人間を上回る得点をたたき出している[4]．このように，深層学習や深層強化学習は今まさに最も注目される分野の 1 つであり，幅広い他の領域での活用が進められている．数値流体力学(CFD)の分野においても，深層学習を用いた Reynolds-Averaged Navier-Stokes (RANS)乱流モデルの研究[5]や流体解析結果の予測に関する研究[6]など，既に適用は始まっている[7]．CFD に限らず，流体力学の分野としてどのような利用法があるか，どのようなパラダイムシフトを起こせるか，さらに模索する必要がある．

本研究では，深層強化学習を流体制御の分野に応用できないかを検討する．例えば，宇宙航空研究開発機構(JAXA)では H3 ロケットなどの開発が進められており，CFD や流体に関係する設計項目は数多い．しかし，一旦設計が完了すれば，仮にさらに良い機体形状が考案できても，なかなか反映が難しい．一方，流体制御デバイスのような後から貼り付けることができるような小型のデバイスは，主要な機能に悪影響を与えないことが保証できれば採用される可能性があり，空力性能の向上や流体振動の低減化を目指すことができる．従って，流体制御デバイスに関しては研究を進めていくことは必要である．そこで，本研究では過去に研究事例の多い円柱カルマン渦のフィードバック制御[8-13]に試行し，深層強化学習の特性を議論するとともに，従来の制御則に基づくフィードバック制御との比較を通してその可能性を調べる．

2　円柱カルマン渦の制御問題

本研究では，主流マッハ数 $M_\infty = 0.05$，レイノルズ数 100 における円柱のカルマン渦放出に起因した揚力変動 $C_{L_{\mathrm{RMS}}}$ の低減化を目的とした流体制御を課題として設定した．レイノルズ数 100 ではカルマン渦は層流であり，円柱方向に流れの 3 次元性が発生しない 2 次元の渦構造となることも知られていることから，深層強化学習を試行する最初の題材として選択した．制御手法としては揚力方向に円柱を振動させたり，回転させたりする手法や，円柱表面の剥離点付近に吸込み/湧き出しを行う制御方法などがある．また，制御側については観測値に対して制御ゲインと時間遅れを与えてフィードバックするやり方が主要[8,9,11]であるが，限られた観測値から支配的な Proper Orthogonal Decomposition (POD)モードを推定して制御したり[10,12]，深層ではない強化学習を用いる手法[13]など，様々な研究事例がある．

本研究では吸込み/湧き出しによる制御方法を選択する．D を円柱直径とすると，Fig. 1 に示すように，円柱

から $0.5D$ 下流における Y 方向速度 V^{mon} を観測し，剥離点付近の円柱後端から±110 度の位置 2 か所に作用点を設ける．作用点に与える吸込み/湧き出し流速 U^{act} はそれぞれ逆位相で与える．比江島らの研究[11]より，ゲイン G と時間遅れ τ を設定し，以下のような制御則で $C_{L_{\mathrm{RMS}}}$ を低減化可能であることが知られている．深層強化学習を利用したフィードバック制御との比較対象として，式(1)に示す制御則を利用したフィードバック制御を利用する．

$$U^{\mathrm{act}} = GV^{\mathrm{mon}}(t-\tau) \tag{1}$$

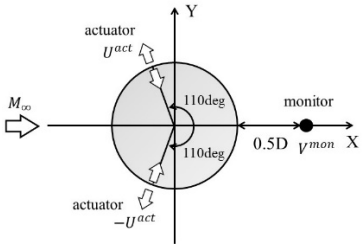


Fig. 1 Setup of feedback control.

3 流体計算手法

JAXA 内製の圧縮性 CFD 解析プログラム upacs-LES[14]を用いた．upacs-LES はマルチブロック構造格子を用いた有限体積法を採用している．支配方程式は層流の Navier-Stokes 方程式であり，対流項の評価はリミッタなしの 2 次精度 MUSCL と SLAU スキーム[15]を，粘性項は 2 次精度中心差分で評価した．時間積分は 3 点後退差分による MFGS 陰解法[16]を利用し，内部反復を 5 回行った．検証として 7700 セルと 29 万セルの 2 種類の格子にて検証した．渦放出周波数(ストローハル数 St)はそれぞれ 0.162，0.165 であり，文献値[17]といずれもよい一致を示す．そこで，本研究では 7700 セルの格子を用いた．フィードバック制御がない場合，$C_{L_{\mathrm{RMS}}}$ は 0.241 となった．

4 深層強化学習の手法

深層強化学習とは強化学習の枠組み[18,19]と深層学習の枠組み[20]からなる．それぞれの手法の詳細は参考文献[4,18-20,22]を参照されたいが，本稿では議論に係る部分のみを抽出する．

強化学習の基本的な仕組みを Fig. 2 に示す．強化学習では，ある環境の中で，あるエージェントが行動している状況を考える．エージェントは，環境から状態と報酬を得て，それを基に行動を選択する．環境は，その行動を受けて状態を変化させ，それに応じた報酬をエージェントに与える．そしてまたエージェントが環境から状態と報酬を得る，ということを繰り返しながら，将来に渡る報酬を最大化する行動を学習する．

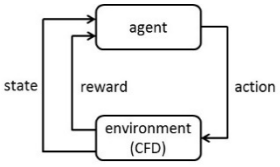


Fig. 2 Schematic of reinforcement learning

強化学習において，状態 $s \in S$，行動 $a \in A$，及び報酬の関係はマルコフ決定過程を用いて表される．状態 s で行動 a をとった際に次の状態 s' へは状態遷移確率 $P(s'|s,a)$ で遷移し，その際に報酬 $r(s,a,s')$ を得るものとする．これら、状態、行動、報酬、状態遷移確率の 4 つの要素によって環境の挙動が定式化される．本研究においては，状態は観測点の Y 方向速度 V^{mon}，報酬は $C_{L_{\mathrm{RMS}}}$ を変数とした後述する関数 R，行動は作用点の速度 U^{act} である．エージェントは政策と呼ばれる(確率的な)関数に基づいて行動を決定するわけだが，強化学習とは報酬が多くなるような政策を，マルコフ決定過程の枠組みの中で見つける問題に帰着する．ただし，ある時刻の短期的な報酬のみではなく，長期的に得られる報酬が大きくなるような政策を選択する必要があることから，下記のように割引報酬和で表した収益 G_t を定義する．

$$G_t = \sum_{\tau=0}^{\infty} \gamma^{\tau} r_{t+1+\tau} \tag{2}$$

ここで，$r_{t+1} = r(s_t, a_t, s_{t+1})$，$\gamma$ は割引率 $(\gamma \in [0,1])$ であり，t は時間を表す．将来の報酬は割引いて考え，現時点の報酬を重視する収益である．

強化学習の代表的な手法である Q 学習では，ある政策 π に対して，次式のように行動価値関数 Q^{π} を導入する．

$$Q^{\pi}(s,a) = \mathbb{E}^{\pi}[G_t | S_t = s, A_t = a] \tag{3}$$

ただし，$\mathbb{E}^{\pi}$ は政策 π に従って行動をとり続けた場合の期待値を表す．先に述べたように，強化学習はできるだけ報酬が多くなる政策を見つける問題となるが，マルコフ決定過程では，他のどんな政策よりも優れているか同等な最適政策 π^* が少なくとも 1 つ存在し，全ての最適政策 π^* は唯一の（最適な）行動価値関数を共有することが知られている．従って，最適行動価値関数 Q^* は以下のように表される．

$$Q^*(s,a) = Q^{\pi^*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \quad (4)$$

そして，Q^* に関しては次のベルマン最適方程式が成立する．

$$Q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\Big[r(s,a,s') + \gamma \max_{a' \in A(s')} Q^*(s',a')\Big] \quad (5)$$

ただし，実問題ではベルマン最適方程式含まれる状態遷移確率 $P(s'|s,a)$，最適行動価値関数 Q^* は未知であることが多い．そこで，標本を十分多く取れば統計的確率は理論的確率に近づくという大数の法則を利用し，式(5)を次式のように近似する．

$$Q^*(s,a) \approx \frac{1}{N}\sum_{n=1}^{N}\Big[r(s,a,s'_n) + \gamma \max_{a' \in A(s'_n)} Q^*(s'_n,a')\Big] \quad (6)$$

ただし，n は反復回数を表し，N は十分大きいとする．Q_n を Q^* の n ステップ目の近似とすると，価値反復法と呼ばれる Q 学習は，反復法を利用して Q_n を収束させることに帰着する．

$$\begin{aligned} Q_n(s,a) &\approx \frac{1}{n}\sum_{i=1}^{n}\Big[r(s,a,s'_i) + \gamma \max_{a' \in A(s'_i)} Q_{i-1}(s'_i,a')\Big] \\ &= \frac{n-1}{n} Q_{n-1}(s,a) \\ &\quad + \frac{1}{n}\Big[r(s,a,s'_n) + \gamma \max_{a' \in A(s'_n)} Q_{n-1}(s'_n,a')\Big] \\ &= Q_{n-1}(s,a) \\ &\quad + \frac{1}{n}\Big[r(s,a,s'_n) + \gamma \max_{a' \in A(s'_n)} Q_{n-1}(s'_n,a') \\ &\qquad - Q_{n-1}(s,a)\Big] \end{aligned} \quad (7)$$

ここで，$1/n$ を $\alpha_n \in (0,1)$ なる α_n で置き換えると，次式を得る．

$$\begin{aligned} Q_n(s,a) = Q_{n-1}(s,a) &+ \alpha_n\Big[r(s,a,s') \\ &+ \gamma \max_{a' \in A(s')} Q_{n-1}(s',a') \\ &- Q_{n-1}(s,a)\Big] \end{aligned} \quad (8)$$

α_n は学習率と呼ばれる係数であり，ある条件を満たせば $\alpha_n = 1/n$ でなくても，Q_n が Q^* に収束することが知られている．s' を n ステップ目の標本過程のデータとし，ベルマン最適方程式の右辺を Q_{n-1} を用いて評価すると以下となる．

$$Q^*(s,a) \approx r(s,a,s') + \gamma \max_{a' \in A(s')} Q_{n-1}(s',a') \quad (9)$$

上式を用いて，式(8)は以下のように書ける．

$$Q_n(s,a) = Q_{n-1}(s,a) + \alpha_n[Q^*(s,a) - Q_{n-1}(s,a)] \quad (10)$$

右辺第 2 項に着目すると，Q^* の予測値である Q_{n-1} に対し，式(9)にて表される Q^* は s, a のもとで実際に s' に遷移して評価した Q^* の(経験的な)値であり，教師信号とみなすことが出来る．つまり，Q 学習とは反復法により予測値 Q_n を教師信号 Q^* に収束させているということが明確になる．なお，式(8)や式(9)の右辺第 2 項は TD 誤差と呼ばれる．

DQN に代表される深層強化学習では，Q^* を以下のようにニューラルネットで関数近似する．

$$Q^*(s,a) \approx Q(s,a;\theta) \quad (11)$$

ここで，θ はニューラルネットに表れる係数の集合を表す．DQN では，θ を得るために，誤差関数 $J(\theta)$ として次式に示すような TD 誤差の二乗を使用する．

$$J(\theta) = \frac{1}{2}\Big[r(s,a,s') + \gamma \max_{a' \in A(s')} Q(s',a';\theta) - Q(s,a;\theta)\Big]^2 \quad (12)$$

そして，勾配降下法：

$$\theta_{n+1} = \theta_n + \alpha \nabla_\theta J(\theta) \quad (13)$$

によって，θ を収束させる．特に，深層学習で用いられる確率的勾配降下法（ミニバッチ学習）を利用する．

Mnih らがテレビゲームで用いた DQN では，上記の学習に加え，学習の安定化や収束性の向上を目指した，以下の ① ～ ③ の工夫が施されており [4]，この工夫を含めて DQN と呼ばれる．また，強化学習一般で用いられる重要な工夫として以下の ④ があり，DQN でも用いられている．

① 体験再生

データが得られた順にニューラルネットを更新すると直近の強い相関を持つ状態に過剰にフィットしてしまい，過去の入力に対する推定精度が悪化し，収束が安定しない．そこで，$(s_t, a_t, s_{t+1}, r_{t+1})$ をバッファに保存し，このバッファからランダムに抽出したものをミニバッチとしてニューラルネットを更新する．バッファから抽出する際，優先順位を付けて抽出する優先順位付き体験再生 [21] も利用される．優先順位付けは TD 誤差や報酬を用いるなど，様々なやり方がある．

② ターゲットネットワーク

式(12)に示すように，TD 誤差項内の $r(s,a,s')+\gamma\max_{a'\in A(s')}Q(s',a';\theta)$ は教師信号であり，予測値 $Q(s,a;\theta)$ を学習させるわけだが，教師信号を同じニューラルネットの係数 θ から求めた場合，係数 θ の更新のたびに教師信号も影響を受け収束が安定しない．そこで，学習用のニューラルネット θ_{train} とは別に教師信号用のターゲットネットワーク θ_{target} を用意し，誤差関数を以下のように変更する．

$$J(\theta_{\mathrm{train}})=\frac{1}{2}\Big[r(s,a,s')+\gamma\max_{a'\in A(s')}Q(s',a';\theta_{\mathrm{target}t})-Q(s,a;\theta_{\mathrm{train}})\Big]^2 \quad (14)$$

θ_{target} は決められた時間ステップごとに，混合率 $\omega\in(0,1]$ と学習した係数 θ_{train} を用いて下記のように更新する．

$$\theta_{\mathrm{target}}=\omega\theta_{\mathrm{train}}+(1-\omega)\theta_{\mathrm{target}} \quad (15)$$

③ 報酬と TD 誤差のクリッピング

報酬の大きさが違うゲームに対しても汎用的に使えるようにするため，報酬 R を $R\in[-1,1]$ となるようにクリッピングし，大きさを揃える．さらに，誤差関数の勾配が大きくなり過ぎないようにするため， TD 誤差を e として，以下のようにクリッピングしたものを誤差関数とする．

$$J(\theta)=\begin{cases}\frac{1}{2}e^2 & e\in[-1,1]\\ e-\frac{1}{2} & \text{otherwise}\end{cases} \quad (16)$$

④ 探索と利用のトレードオフ

強化学習において本質的な問題である探索と利用のトレードオフに対し，確率 ε でランダムに行動を選択し，$1-\varepsilon$ で行動 $a=\mathrm{argmax}_a Q^*(s,a)$ をとる ε-greedy 法が一般的に利用される．なお，ε は学習が進むにつれて徐々に小さくする．

ここまで紹介した Q 学習は，連続的な行動を取り扱うことは困難であるというデメリットがある．本研究における行動 a は作用点の速度 U^{act} であり，連続的に表現する方が素直であると考えられる．そこで，本研究では連続的な行動の取扱いも可能にするため，政策を決定的な関数としてニューラルネットで近似する Deep Deterministic Policy Gradient (DDPG)[18,22] を採用した．DDPG のネットワーク構成を Fig. 3 に示す．DDPG では Q 関数を近似する DQN と同様の Critic ネットワーク $Q(s,a;\theta^Q)$ と，決定論的政策を近似する Actor ネットワーク $\mu(s;\theta^\mu)$ の 2 つから成る．Critic ネットワークは，DQN と同様の誤差関数を用いることにより学習が可能である．Actor ネットワークについては，反復法で θ^μ 求めるために、まず目的関数を開始時点 $t=0$ からの割引報酬和とする．

$$J(\mu_{\theta^\mu})=\mathbb{E}[G_0]=\mathbb{E}_s\big[\mathbb{E}^\mu[G_0|s]\big]=\mathbb{E}_s\big[Q\big(s,\mu(s)\big)\big] \quad (17)$$

そして，θ^μ も Q 関数と同様に，以下のようにミニバッチ確率的勾配降下法によって収束解を求める．

$$\theta^\mu_{n+1}=\theta^\mu_n+\alpha\frac{\partial J}{\partial\theta^\mu} \quad (18)$$

なお，α は式(8)等に出てくる学習率であるが全て同じ値とは限らない．ここで，式(17)より，式(18)の右辺第 2 項は以下のように近似され，さらに，chain rule を使って変形することが出来る．

$$\begin{aligned}\frac{\partial J}{\partial\theta^\mu}&\approx\mathbb{E}_s\left[\frac{\partial Q(s,\mu(s);\theta^Q)}{\partial\theta^\mu}\right]\\&=\mathbb{E}_s\left[\frac{\partial Q(s,a;\theta^Q)}{\partial a}\frac{\partial\mu(s;\theta^\mu)}{\partial\theta^\mu}\right]\\&\approx\frac{1}{N}\sum_s\left[\frac{\partial Q(s,a;\theta^Q)}{\partial a}\frac{\partial\mu(s;\theta^\mu)}{\partial\theta^\mu}\right]\end{aligned} \quad (19)$$

なお，N はミニバッチ数を表し，右辺をミニバッチの平均で近似している．右辺第 1 項は Critic ネットワークで関数近似された行動価値関数を行動で微分したもので，右辺第 2 項は Actor ネットワークで関数近似された決定論的政策を Actor ネットワークの係数で微分したものである．

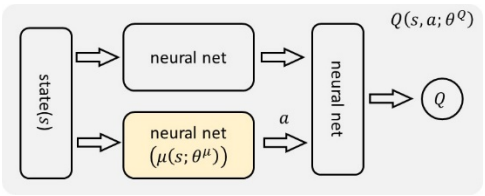


Fig. 3 Neural networks for DDPG

DDPG においても学習を安定化させ収束性を向上させるために，体験再生，ターゲットネットワークなど DQN と同様のテクニックを利用するが，探索のアルゴリズムとして行動に一様乱数を使うのではなく，時間相関のある Ornstein-Uhlenbeck 過程を利用する．離散化した Ornstein-Uhlenbeck 過程 χ_i は以下のように表される．

$$\chi_{i+1}=\chi_i+\theta(\mu-\chi_i)dt+\sigma\sqrt{dt}\mathcal{N}(0,1) \quad (20)$$

なお，$\mathcal{N}(0,1)$ は平均が 0，分散が 1 の正規分布に従う

乱数である．

本研究で対象とする円柱カルマン渦のフィードバック制御に適用する場合，観測点で得られる状態 V^{mon} と作用点に入力する行動 U^{act} には時間遅れがある．行動が時間遅れをもって作用する系では時間遅れのあるマルコフ決定過程[23]として取り扱う必要がある．時間遅れを τ とすると，$n_a = \tau/\Delta t$ 個の行動 U^{act} を状態に加える必要がある．一方，観測点の V^{mon} に関してもシステムの状態を表すためには時間方向に十分な点数 n_s が必要となる．まとめると，本研究における状態 s は以下のようになる．

$$s = \left[V_t^{\mathrm{mon}}, V_{t-\Delta t}^{\mathrm{mon}}, \cdots, V_{t-(n_s-1)\Delta t}^{\mathrm{mon}}, U_t^{\mathrm{act}}, U_{t-\Delta t}^{\mathrm{act}}, \cdots, U_{t-(n_a-1)\Delta t}^{\mathrm{act}}\right] \quad (21)$$

本研究ではpython3 系と tensorflow1.2.1 を用いて深層強化学習プログラムを開発した．

5　制御則によるフィードバック制御

本章では式(1)を用いた制御を行い，最大で得られる $C_{L_{\mathrm{RMS}}}$ の低減化量について調べた．始めに，$G = 1.0$ と固定し，時間遅れ τ の変化が $C_{L_{\mathrm{RMS}}}$ の低減化に及ぼす影響を調べた結果を Fig. 4 に示す．T をカルマン渦放出周期とすると，$\tau/T = 0.4$ で極小値が得られることが分かる．ただし，制御が進むと T は変化することが知られていることから [8]，比江島ら [11]と同様に，観測点の V^{mon} が正から負へと変わる瞬間を基準時刻とし，この基準時刻と 1 つ前の基準時刻の差を T とした．次に，極小値が得られた時間遅れ($\tau/T = 0.4$) においてゲインの影響を調べた．Figure 4 に示すように，$G = 2.0$ で $C_{L_{\mathrm{RMS}}}$ の最小値(0.037)が得られた．制御無し($C_{L_{\mathrm{RMS}}} = 0.241$)と比べ，85%程度の低下が得られた．Figure 5 に円柱にかかる時系列 C_L 分布と作用点の吸い込み/湧き出し速度 U^{act} を示す．t = 243 から制御を開始する．制御開始直後，及び周期が更新されるタイミングで一時的に U^{act} が不連続に変化し，その結果 C_L にスパイクが現れるが，制御が定常状態に落ち着くと C_L, U^{act} ともに sin 波に相当する波形を示す．

6　深層強化学習によるフィードバック制御

DDPG を用いたフィードバック制御を試みる．4 章で紹介したように，DDPG は深層学習と強化学習のそれぞれに要するパラメータが沢山あり，列挙すると以下の通りである．

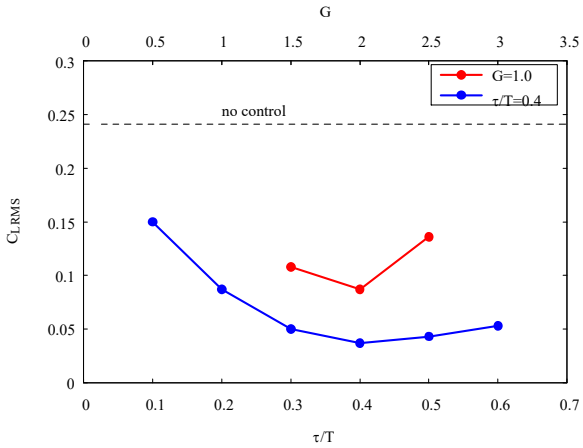


Fig. 4　Effect of G and τ on $C_{L_{\mathrm{RMS}}}$

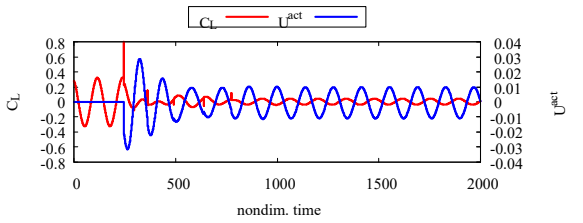


Fig. 5　Result of feedback control based on control law.

- ✓ 入力層
- ✓ 行動の大きさ U_0^{act}
- ✓ Actor ネットワークの学習率
- ✓ Critic ネットワークの学習率
- ✓ 割引率
- ✓ エピソード数 N_{max} とエージェントが環境に作用する時間ステップ
- ✓ 深層学習のハイパーパラメータ(隠れ層，ノード数，活性化関数，ミニバッチ数，正規化層の有無)
- ✓ ターゲットネットワークの混合率 ω と θ_{target} の更新間隔
- ✓ 体験再生におけるバッファサイズ
- ✓ 報酬関数
- ✓ 優先順位の有無とその対象とする関数
- ✓ 探査アルゴリズム

上記すべてのパラメータに関する感度を本研究だけで調べるわけにはいかない．それぞれに対して，決め打ちで設定したパラメータについてはその設定値を，感度を調べたパラメータについてはその影響について議論する．

6.1　入力層

制御がない場合のカルマン渦放出周波数は $St =$

0.162 (無次元周期≈ 120)である．CFD の無次元時間刻み幅は 0.02 であり，1 周期のカルマン渦の解析には約 6000 ステップが必要である．ここで，深層強化学習のエージェントは 100 ステップごとに環境 (CFD) から状態と報酬を得て行動を返すとした．観測点の V^{mon} は無次元時間で $\Delta t = 24$ ごとに $n_s = 6$ 個の状態を抽出した．4 章で議論したように，時間遅れのあるマルコフ決定過程を考える必要があり，無次元時間で $\Delta t = 2$ ごとに 24 点の作用点の U^{act} も入力とした．従って，状態 s は下記のように 30 次元のベクトルで表される．

$$[V_t^{\mathrm{mon}}, V_{t-24}^{\mathrm{mon}}, \cdots, V_{t-120}^{\mathrm{mon}}, U_t^{\mathrm{act}}, U_{t-2}^{\mathrm{act}}, \cdots, U_{t-46}^{\mathrm{act}}] \quad (22)$$

6.2 行動の大きさ，学習率，割引率

行動の大きさ，Actor/Critic ネットワークの学習率，及び割引率を Table 1 に示す．行動 U^{act} は $U^{\mathrm{act}} \in [-0.04, 0.04]$ とした．その他のパラメータは一般的に使われる値を設定した．

Table 1 List of miscellaneous parameters.

行動の大きさ(U_0^{act})	0.04
Actor ネットワークの学習率	0.0001
Critic ネットワークの学習率	0.001
割引率	0.99

6.3 エピソード数と時間ステップ

エピソード数 $N_{\max}$ は 300 とした．エージェントが環境に作用する時間ステップは，前述の通り，CFD の 100 ステップ(Δt=2)ごととした．制御がない場合のカルマン渦放出周期は約 120 であることから，カルマン渦 1 周期に対して 60 回アクチュエータが作用する．1 エピソード辺り CFD は無次元時間で 700 の計算を実施しており，制御無しの場合にはカルマン渦が 6 回程度放出される時間幅である．

6.4 深層学習のハイパーパラメータ

本研究で用いたベースラインとなるネットワークを Fig. 6 に示す．Actor ネットワーク θ^{μ} の入力層は式(22)に示した 30 ノードであり，出力は行動 a の 1 ノードである．隠れ層は第 1 層(fc1)が 400 ノードの全結合層，第 2 層(fc2)が 300 ノードの全結合層である．活性化関数は出力層が tanh 関数，それ以外は Rectified Linear Unit (ReLU)である．Critic ネットワーク θ^{Q} も同様に 400 ノード，300 ノードの全結合層が隠れ層としてあり，それぞれ ReLU を活性化関数とした．ただし，出力層では活性化関数を用いず，また正規化層も用いていない．本研究ではミニバッチ数は 32 と固定した．

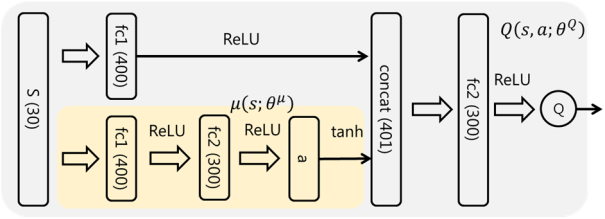


Fig. 6 Layout of actor and critic networks.

6.5 ターゲットネットワーク

本研究では，θ_{target} は毎時間ステップごとに，式(15)で θ_{train} とブレンドした．なお，混合率 ω は 0.0001 とした．

6.6 体験再生のバッファサイズ

バッファサイズは 20,000 とした．そしてランダムに 32 セットのデータをミニバッチとして抽出した．

6.7 報酬関数

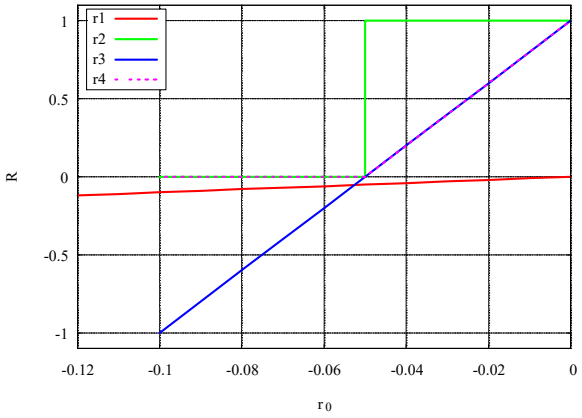


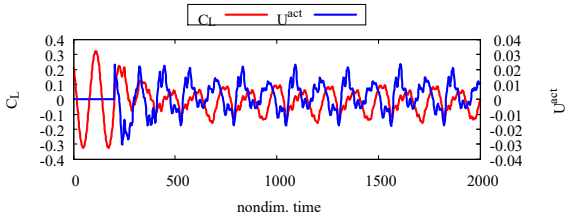
Fig. 7 Reward clipping functions.

報酬は次式のように設定する．

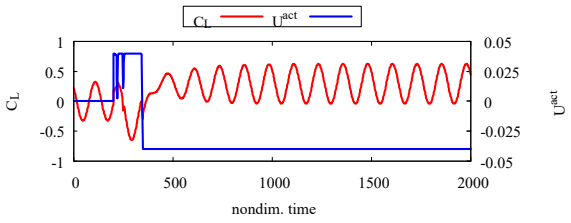
$$r_0 = -\left(C_{L_{\mathrm{RMS}}}\right)^2 - \mathrm{c}_1(U^{\mathrm{act}})^2 - \mathrm{c}_2(\Delta U^{\mathrm{act}})^2 \quad (23)$$

ここで，$\Delta U^{\mathrm{act}} = U_{t+1}^{\mathrm{act}} - U_t^{\mathrm{act}}$ であり，前の時間ステップの行動との差分値である．本研究の目的は $C_{L_{\mathrm{RMS}}}$ の最小化であるため，$-\left(C_{L_{\mathrm{RMS}}}\right)^2$ を報酬とし最大化すればよいが，併せて $-\left(C_{L_{\mathrm{RMS}}}\right)^2$ だけではなく，できるだけ作用点の入力速度 U^{act} が小さいことが望ましいことから，右辺第 2 項の $-(U^{\mathrm{act}})^2$ をペナルティとして報酬に加えている．更に，作用点の速度変化が激しいことは物理的に望ましくないことから，$-(\Delta U^{\mathrm{act}})^2$ もペナルティとして報酬に加えた．右辺にある 3 項のバランス

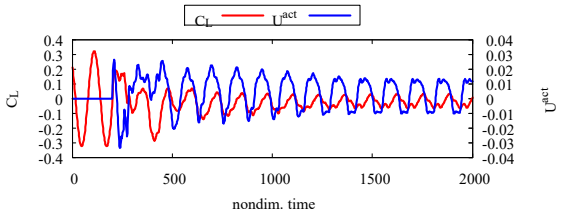
を考慮し，$c_1 = 10.0, c_2 = 100.0$ とした．そして，r_0 を変数とし，Fig. 7 に示す 4 つの関数 $r1, r2, r3, r4$ を用いて報酬 R に変換した．



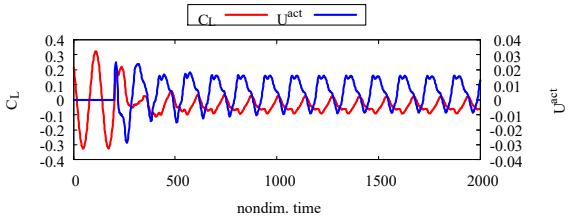
(a)　$r1$.



(b)　$r2$.



(c)　$r3$.



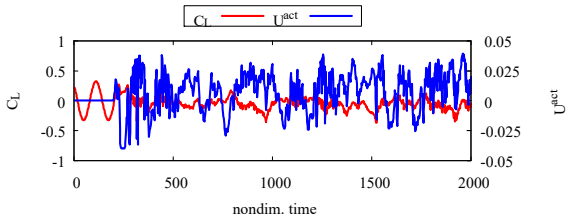
(d)　$r4$.

Fig. 8　Effect of reward functions.

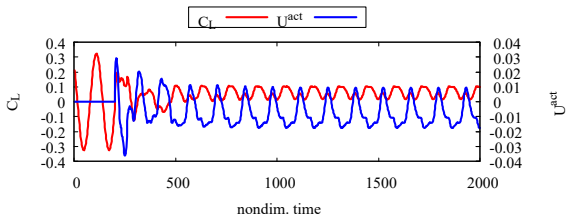
まず，$r1$ は $R = r1(r0) = r0$ であるが，$r2, r3, r4$ は Fig. 7 に示すように，$(r_0, R) = (0.05, 0)$ (制御無し時の $C_{L\,\mathrm{RMS}}^2 \approx 0.05$) を通り，-1～1，もしくは 0～1 を連続的か離散的に取る．なお，$r2, r3, r4$ は $r_0 < -0.1$ となると強制終了し，次のエピソードへと進むようにした．これら 4 つの関数 $r1, r2, r3, r4$ を用いた結果を Fig. 8 に示す．いずれも $t = 200$ から制御を開始する．Figure 8 より，報酬関数によって結果が大きく異なることが分かる．$r1$ では Fig. 5 の従来の制御則の結果と比べて U^{act} は高周波の振動成分を含み，その結果，C_L も同様に高周波の振動が見られる．$r2$ では，U^{act} は下限の-0.04 で一定になってしまっており，C_L に関しても DC 成分を持った振動となる．$r3, r4$ は周期的な U^{act} が得られており，C_L に関しても同様である．次に得られた $C_{L\,\mathrm{RMS}}$ を Fig. 10 にて比較する．なお，$C_{L\,\mathrm{RMS}}$ は C_L の振幅がほぼ定常とみなせる $t = 1{,}200 \sim 2{,}000$ で算出した．$r2$ では制御無しの結果とほぼ同様であり，$r1, r3, r4$ を比較すると $r3$ の報酬関数を用いた場合が最もよい結果となった．そこで，以降の議論では $R = r3(r_0)$ とする．

6.8　優先順位付け

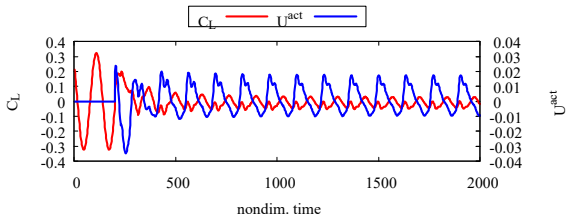
体験再生で利用するバッファの優先順位付けについて，対象とする関数を P とすると，優先度を $(P + \varepsilon)^{\alpha}$ でつけた．ここで，$\varepsilon = 10^{-6}, \alpha = 0.6$ とし，P は TD 誤差と，報酬関数を用いた $P = r3 + 1$ と $P = |r3|$ の計 3 通りを試行した．



(a)　TD error.



(b)　Reward $(r3 + 1)$.



(c)　Absolute value of reward($|r3|$).

Fig. 9　Effect of prioritized replay using $r3$ for reward function.

C_L, U^{act} の時系列変化を Fig. 9 にて比較する．テレビゲームを対象とした DQN で用いられているように TD 誤差を優先度の関数として利用した場合，U^{act} には周

期性がみられずほぼランダムな入力となっている．一方，$P = r3 + 1, |r3|$ を用いると U^{act} はのこぎり波のようになる．C_L をみると，サイン波ではなく大小 2 つのピークを持った結果となった．C_L の振幅がほぼ定常とみなせる $t = 1{,}200\sim2{,}000$ で算出した $C_{L_{\mathrm{RMS}}}$ を Fig. 10 にて比較する．$P = |r3|$ が最もよい性能を示し，$C_{L_{\mathrm{RMS}}} = 0.024$ となった．

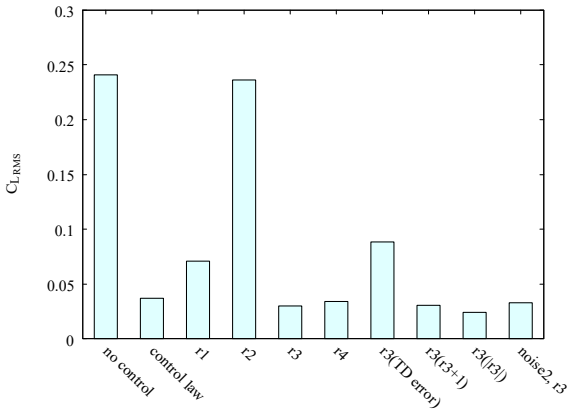


Fig. 10 Comparison of $C_{L_{RMS}}$

なお，報酬関数は $R = r3(r_0)$ とし，$P = |r3|$ とした最もよい C_L の低減化が得られたケースにおける Q 関数の時系列分布を Fig. 11 に示す．先述した通り，本研究では 1 エピソード辺り 35,000 ステップの CFD 解析を 300 エピソード実施した．Figure 11 を見ると，本解析の範囲内でほぼ収束している．計算は Intel® Xeon® CPU E5-2698 v3 を 2 つ搭載した合計 32 コアのサーバを用いて実施した．解析時間は全体で約 2 日を要し，そのうちの約 91%は CFD に要した．

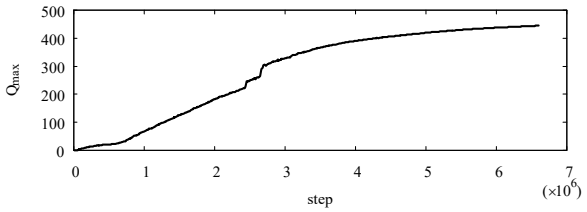


Fig. 11 Convergence history of Q^*

6.9 探索アルゴリズム

探索のためのノイズ χ は，式(20)に示す離散化された Ornstein-Uhlenbeck 過程を利用した．パラメータとなる θ, σ, dt については，$\theta = 0.15$，$\sigma = 0.2$，$dt = 1.0$ と $\theta = 0.15$，$\sigma = 0.4$，$dt = 0.04$ の 2 つ条件の結果の一例を Fig. 12 にて比較する．前者は上記の議論で用いたパラメータセットで，Lillicrap らが利用する値 [22]である．一方，Fig. 12 より，$\theta = 0.15$，$\sigma = 0.4$，$dt = 0.04$ のノイズは前者に比べて高周波成分が抑制されたノイズであることが分かる．探査ノイズは，次式に示すように $\mu(s;\theta^{\mu})$ と混合し，エピソード数 N が進むにつれて探査ノイズが徐々に小さくなるようにした．

$$a = (1 - 0.98^N) \times \mu(s;\theta^{\mu}) + 0.98^N \times U_0^{\mathrm{act}}\chi \quad (24)$$

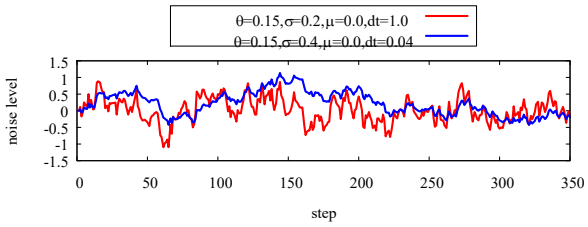


Fig. 12 Noise based on Ornstein-Uhlenbeck process.

ここまでで最も C_L の低減化性能が良かった Fig. 8 (c) ($R = r3(r_0)$) や Fig. 9 (c) ($R = r3(r_0)$，$P = |r3|$) を見ると，U^{act} や C_L 分布は細かい振動成分を有しており，$\theta = 0.15$，$\sigma = 0.4$，$dt = 0.04$ のノイズ(以降，noise2 と呼ぶ)を利用すればこの細かい振動を除去できるのではないかと考えられる．そこで，報酬関数を $R = r3(r_0)$ とし，体験再生の優先順位付けをしないもの，$P = |r3|$ で順位付けする 2 ケースの解析を実施した．優先順位付けをしないケースの結果を Fig. 13 に示す．

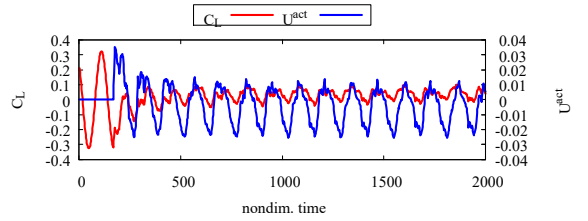


Fig. 13 Effect of noise.$R = r3(r_0)$ without prioritized.

Figure 8 (c)と比較すると，ノイズを変更したからといって結果から高周波の振動がなくなるわけではない．Figure 10 にて $C_{L_{\mathrm{RMS}}}$ を比較すると，Lillicrap らが利用したノイズでは $C_{L_{\mathrm{RMS}}} = 0.030$，noise2 では $C_{L_{\mathrm{RMS}}} = 0.033$ と若干ではあるが結果は悪化した．次に $P = |r3|$ で順位付けした場合に noise2 を適用した結果，$r_0 < -0.1$ となるエピソードが多発し，学習を進めることができなかった．このように，結果を改善するパラメータ設定の組み合わせは，別のパラメータを変更した場合に必ずしも同様の傾向を示すわけではない．本研究で調べたパラメータは，すべてニューラルネットの係数

を安定的によい収束解を求めるようにすることを目的としている．現段階では個々のパラメータがニューラルネットの膨大な係数に与える影響を調べ切れてはいないため，別のパラメータを変えた場合に，これまでよかった設定の組み合わせがうまくいかなくなるといった現象が起きている．今後は，よりニューラルネットの膨大な係数の動きを調べていく必要がある．

7　従来制御則と深層強化学習から得られた結果の比較

従来制御則においてもっとも性能が良かった $G = 2.0, \tau = 0.4$ の条件の結果，及び深層強化学習で最も性能が良かった報酬関数を $R = r3(r_0)$，$P = |r3|$ で優先順位付けをした体験再生を利用した結果を比較する．それぞれ $C_{L_{\mathrm{RMS}}}$ は0.037，0.024である．制御無しの場合，$C_{L_{\mathrm{RMS}}}$ は0.241であり，深層強化学習を利用すると1/10程度まで低下した．また，従来の制御則を利用した場合と比べ，34%ほど $C_{L_{\mathrm{RMS}}}$ は低下しており，DDPGによる深層強化学習により従来制御則を超えることができた．本項ではこれらの結果を比較する．まず U^{act} をFig. 14にて比較する．従来制御則ではほぼsin波となっているが，DDPGではのこぎり波のような波形となり，また $U^{\mathrm{act}} = 0$ に関して対称ではない．また，極大値は $U^{\mathrm{act}} = 0.018, 0.012$ の2か所に存在し，極小値は1つで-0.01である．一方，従来制御則では極大値，極小値ともに1つで，それぞれ0.010, -0.011である．周期に関しても，従来制御則は平均で $T = 138.4$ であるのに対しDDPGでは $T = 134.5$ と短く，Fig. 14から山の位置がずれる様子が観察される．

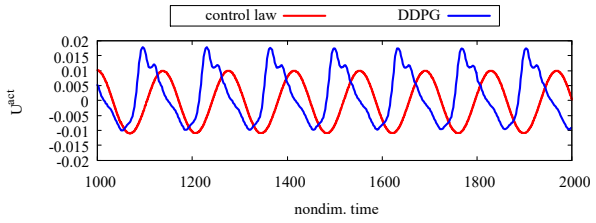


Fig. 14　Comparison of U^{act} plot between result of control law and DDPG.

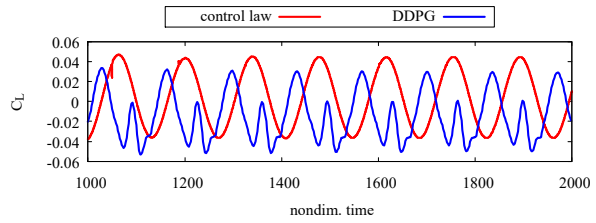


Fig. 15　Comparison of C_L plot between result of control law and DDPG.

Figure 15に示した C_L の分布についてみると，従来制御則では U^{act} と同様にほぼsin波になっており，極大値，極小値はいずれも1つで，それぞれ0.047, -0.038である．一方，DDPGでは平均 $T = 134.5$ の周期の中に山/谷が2つ現れる．極大値は0.034, -0.001，極小値は-0.053, -0.047である．深層強化学習では U^{act} が2つの極大値を持つのこぎり波の波形になり，C_L が1周期に山/谷を2つ持つようになった理由と，なぜこのような波形が $C_{L_{\mathrm{RMS}}}$ の低下につながったかについては今後詳細な解析を実施する．

8　まとめ

深層強化学習の1つであるDeep Deterministic Policy Gradient (DDPG)を用いて，レイノルズ数が100の円柱カルマン渦に起因した円柱にかかる揚力のRMS値 $C_{L_{\mathrm{RMS}}}$ を低減するフィードバック制御を試行した．円柱直径の0.5倍だけ円柱下流位置の速度をモニタし，円柱表面2か所からジェットを吸込み/湧き出しさせて $C_{L_{\mathrm{RMS}}}$ をフィードバック制御させた．制御無しの $C_{L_{\mathrm{RMS}}}$ と比較し，DDPGでは約1/10に低下した．また，ゲインと時間遅れの2つをパラメータとする制御則に基づいた従来のフィードバック制御と比較し，DDPGは34%ほど $C_{L_{\mathrm{RMS}}}$ を低減化させることができた．ジェットの吸込み/湧き出し速度は従来制御則ではsin波に近い形状となるが，DDPGでは極大値を2つ持つのこぎり波のような波形となった．C_L 分布については，従来制御則では同様にsin波のような波形となるが，DDPGでは極大値・極小値を2つ持つような特異な形状となった．

上記のように，DDPGを試行することによって従来の制御則に基づくフィードバック制御を超える性能を持ち，今後の可能性が期待できるものであることを示すことができた．しかし，本研究で対象としたレイノルズ数が100の円柱カルマン渦は層流の2次元渦構造となることが分かっている．実用に向けては，乱流といった非線形性が強く表れる流れ場に対しても試行する必要がある．一方，結果を改善するパラメータの組み合わせは，別のパラメータを変更した場合に必ずしも同様の傾向を示すわけではないことが分かった．本解析では，個々のパラメータがニューラルネットの膨大な係数に与える影響を調べ切れてはいないため，今後はニューラルネットの係数について調べる必要がある．最後に，本研究では環境としてCFDを利用しており，Intel® Xeon® CPUを2つもつ計32コアのサーバを用いて2日間を要した．そして，そのうちの91%程度はCFD解析であった．CFDだ

けではなく，実験も併用するなど，効率的な学習を考える必要がある．

引 用 文 献

1) Hinton, G..E., Deng, L, Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., and Vanhoucke, V.: Deep Neural networks for acoustic modeling in speech recognition: The Shared Views of Four Research Groups, *IEEE Sig. Proc. Mag.*, 29 (2012) 82-97.
2) Krizhevsky, A. Sutskever, I, and Hinton, G.E.: ImageNet classification with deep convolution neural networks, *Adv. Neur. Info. Proc. Sys. (NIPS)*, 25 (2012) 1097-1105.
3) Le, Q.V., Ranzato, M.A., Monda, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., and Ng., A.Y.: Building high-level features using large scale unsupervised learning, *Proc. 29th Int. Conf. Mach. Learn.,* (2012).
4) Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, 518, (2015) 529-533.
5) Ling, J., Kurzawski, A., and Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, *J. Fluid Mech.*, 807 (2016) 155-166.
6) 増田正人，中村靖，田村善昭: 深層学習を用いた流体解析結果予測, 第 30 回数値流体力学シンポジウム, F07-3 (2015).
7) Kutz, J. N.: Deep learning in fluid dynamics, *J. Fluid Mech.*, 814 (2017) 1-4.
8) Roussopoulos, K.: Feedback control of vortex shedding at low Reynolds numbers, *J. Fluid Mech.*, 248 (1993) 267-296.
9) Park, S.D., Ladd, D.M., and Hendricks, E.W.: Feedback control of von Karman vortex shedding behind a circular cylinder at low Reynolds numbers, *Phys. Fluids*, 6 (1994) 2390-2405.
10) Gillies, E.A.: Low-dimensional control of the circular cylinder wake, *J. Fluid Mech..*, 371 (1998), 157-178.
11) 比江島慎二，渡邊恭，野村卓史: 流速撹乱による円柱カルマン渦のフィードバック制御, 応用力学論文集, 7 (2004) 1125-1132.
12) Siegel, S., Cohen, K., and McLaughlin, T.: Numerical simulations of a feedback-controlled circular cylinder wake, *AIAA J.*, 44 (2006) 1266-1276.
13) Pivot, C., Mathelin, L., Cordier, L., Guent, F., and Noack, B.R.: A continuous reinforcement learning strategy for closed-loop control in fluid dynamics, *AIAA Paper* 2017-3566 (2017).
14) 榎本俊治，野崎理，今村太郎，山本一臣: LES による円形ジェットの乱流混合騒音の数値予測, 第 21 回数値流体力学シンポジウム, B1-1 (2007).
15) Shima, E., and Kitamura, K.: Parameter-free simple low-dissipation AUSM-family scheme for all speeds, *AIAA J.*, 49 (2011) 584-590.
16) 嶋英志: 構造/非構造格子 CFD のための簡単な陰解法, 第 29 回流体力学講演会講演集 (1997) 325-328.
17) Roshko, A.: On the development of turbulent wakes from vortex streets, *NACA Rep.* 1191 (1954).
18) Silver, D., Lever, G., Heess, AN., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms, *Proc. 31st Int. Conf. Mach. Learn.,* (2014) 387–395.
19) 牧野貴樹，澁谷長史，白川真一: これからの強化学習, (森北出版, 2016).
20) 岡谷貴之:深層学習 (講談社, 2015).
21) Schual, T., Quan, J., Antonoglou, I., and Silver, D.: Prioritized experience replay, *Int. Conf. Learn. Represent.* (2016) arXiv:1511.05952v4.
22) Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning, *Int. Conf. Learn. Represent.* (2016) arXiv:1509.02971v5.
23) Katsikopoulos, K.V., Engelbrecht, S.E.: Markov decision processes with delays and asynchronous cost collections, *IEEE Transo. Autom. Control*, 48 (2003) 568-574.