Vol.2021-DPS-187 No.14
Vol.2021-MBL-99 No.14
Vol.2021-ITS-85 No.14
2021/5/27

近隣車情報を用いた強化学習による自動運転制御 ～報酬付与法の検討と制御挙動の分析～

原田 智広[†1,a)]　服部 聖彦[†2,b)]

概要：近年，AI 技術やセンサ性能の向上に伴い，自動運転の研究が盛んに行われている．本研究では，複数車が同時に走行する環境を想定し，車々間での通信によって能動的に取得される近接車情報を考慮した自動運転制御を考える．具体的には，機械学習の一つである強化学習を用いた自動制御の獲得を目指す．本研究では，協調的な自動制御を実現するための報酬付与法の検討と，強化学習によって得られた制御の挙動を分析する．

キーワード：自動運転，強化学習，車々間通信，報酬付与, Proximal Policy Optimization

Autonomous Driving Control by Reinforcement Learning Using Neighboring Vehicle Information — A Study on Reward Design and Behavior Analysis —

Tomohiro Harada[†1,a)]　Kiyohiko Hattori[†2,b)]

Abstract: In recent years, research on automatic driving has been actively conducted with the improvement of AI technology and sensor performance. In this study, we consider automatic driving control that uses actively acquired information through communication between neighbor vehicles in an environment where multiple vehicles are driving simultaneously. Specifically, we aim to obtain automatic control using deep reinforcement learning, a machine learning method. This study investigates rewarding methods for achieving cooperative automatic control and analyzes the behavior of the control obtained by reinforcement learning.

Keywords: Autonomous driving, reinforcement learning, vehicle-to-vehicle communication, reward design, proximal policy optimization

1. はじめに

近年，AI 技術やセンサ性能の向上に伴い，自動運転の研究が盛んに行われている．特に，自動運転車両の運転制御獲得のために，深層学習を含む機械学習手法が用いられる．本研究では，複数の自動運転車が同一道路を走行する環境を想定し，車々間での通信によって能動的に取得される近接車情報を考慮した自動運転制御を考える．

本研究では，機械学習の一つである強化学習 [7] を用いた自動運転の獲得を目指す．強化学習は，環境状態の知覚とその状態に対する行動制御のインタラクションを通じて環境から得られる報酬に基づいて最適行動を学習する．本研究では特に，車両に搭載された距離センサの情報に加えて，近接車との通信によって得られる他車の制御情報を行動決定のための入力状態として扱う．

近接車情報を用いた強化学習によって複数車が走行する環境での制御を獲得可能か検証するために，本研究ではシミュレーション実験を実施する．シミュレーション環境の

[†1] 現在，東京都立大学，システムデザイン学部
Presently with Faculty of System Design, Tokyo Metropolitan University, Hino, Tokyo 191–0065, Japan
[†2] 現在，東京工科大学，コンピュータサイエンス学部
Presently with School of Computer Science, Tokyo University of Technology, Hatchioji, Tokyo 192–0982, Japan
[a)] harada@tmu.ac.jp
[b)] hattorikh@stf.teu.ac.jp

Vol.2021-DPS-187 No.14
Vol.2021-MBL-99 No.14
Vol.2021-ITS-85 No.14
2021/5/27

構築には，3 次元ゲームやシミュレーションの開発に広く用いられている Unity[3] を使用する．強化学習法としては，近年高い性能が示されている深層強化学習アルゴリズム [1] の一つである Proximal Policy Optimization（PPO）[5] を用いる．

2. 関連研究

複数車両の協調行動を深層学習を用いて獲得する方法として小川らの研究がある．文献 [8] では，深層強化学習の一つである Deep Q-Network（DQN）を用いてラウンドアバウトを含むコース上での自動運転制御を調整する方法を提案している．コンピュータシミュレーションではなく，ラジオコントロールカー（RC カー）を用いた現実環境上で実験を行った．同じく小川らの研究 [9] では，RC カーの複数車の協調行動を実現するための報酬設計とカリキュラム学習の導入を提案した．

Pal らは自動運転アルゴリズムに交通ルールをハードコーディングする代わりに，マルチエージェント環境を設計することで，交通流を最大化する交通ルールを学習させる実験を行った [4]．具体的には，交差点におけるすれ違い制御を，強化学習アルゴリズムの Prosimal Policy Optimization[5] を用いて学習した．他のエージェントと衝突することなく，できるだけ早く目的地に到達したエージェントに報酬を与える．実験の結果，自動運転エージェントが車線や信号の概念を共有することで，センサ入力にノイズが多い場合でも目的地に到達可能であることを示した．

自動運転車の情報共有について, 佐藤らの研究がある [10]．複数の車両，あるいは道路に設置されたセンサ情報を統合的に管理する LDM（Local Dynamic Map）を用いて車両間で情報共有することで協調運転を実現している．

3. 学習環境

本研究では，強化学習を用いて複数車の自動運転制御を学習するシミュレーション実験を行う．シミュレーション環境は，3 次元ゲームやシミュレーションの開発に広く用いられている Unity[2] を用いる．シミュレーション環境は，Adam Streck が公開した学習フレームワーク [6] を独自に拡張して用いる．以降，はじめに本研究で使用する深層強化学習アルゴリズムの Proximal Policy Optimization (PPO)[5] について概説する．次に，各車両で取得可能なセンサ情報と制御方法を説明する．最後に，強化学習で用いる報酬設計を示す．

3.1 Proximal Policy Optimization

強化学習法としては，近年高い性能が示されている深層強化学習アルゴリズムの一つである Proximal Policy Optimization (PPO)[5] を用いる．PPO は Actor-Critic 型の深層強化学習であり，方策を表現する方策ネットワーク（Actor）と，状態価値を表現する価値ネットワーク（Critic）の 2 種類のニューラルネットワークからなる．パラメータ θ によって決定される方策ネットワークを $\pi_\theta(a \mid s)$，価値ネットワークを $V_\theta(s)$ と表す．ここで，s は知覚した状態，a は行動（制御）を表す．このとき，PPO では，以下に示す目的関数を最小化するパラメータ θ を学習する：

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}\left[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right] \tag{1}$$

ここで，$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}}$ であり，$\pi_{\theta_{\mathrm{old}}}$ は更新前のパラメータに基づく方策を表す．$\hat{A}_t$ はアドバンテージ関数と呼ばれ，行動価値関数 $Q^{\pi_\theta}(s,a)$ と状態価値関数 $V^{\pi_\theta}(s)$ から $A_t = Q^{\pi_\theta}(s_t,a_t) - V^{\pi_\theta}(s_t)$ で求める．clip 関数は，$r_t(\theta)$ の値を $1-\epsilon$ から $1+\epsilon$ の間の値に収める．ϵ はユーザパラメータである．PPO は，一度の更新で方策が大きく変動することを抑制することで，安定した学習を実現している．

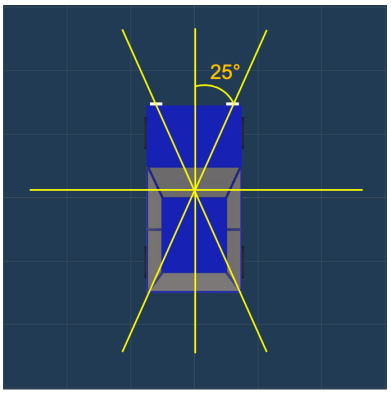

図 1 センサ方向
Fig. 1 Sensor directions

3.2 入力情報

本研究では，図 1 に示すように，前方 3 方向，側方 2 方向，後方 3 方向の計 8 台の距離センサを有する車両を想定する．距離センサは壁や他車との距離を測位する．また，距離センサ内に他車が存在する場合は，その車の速度とハンドル角を隣接車情報として取得する．

本研究で使用するコースは図 2 に示すような複数の路面タイルで構成される．図中の赤枠で囲われた部分が一つのタイルを表し，矢印は進行方向を表す制御マーカである．各車両は，直下に設置された制御マーカが示す方向を取得し，車両と進行方向のなす角度を算出する．

ニューラルネットワークへの入力情報を表 1 に示す．ニューラルネットワークには，自車に関する情報とセンサから得られる情報を入力する．自車情報としては，路面タイルから取得される進行方向と車両方向とのなす角度，自車の最大速度 v_{max}，自車の最大ハンドル角（θ_{max}）を用いる（詳細は後述）．他車情報としては，センサごとに計測

Vol.2021-DPS-187 No.14
Vol.2021-MBL-99 No.14
Vol.2021-ITS-85 No.14
2021/5/27

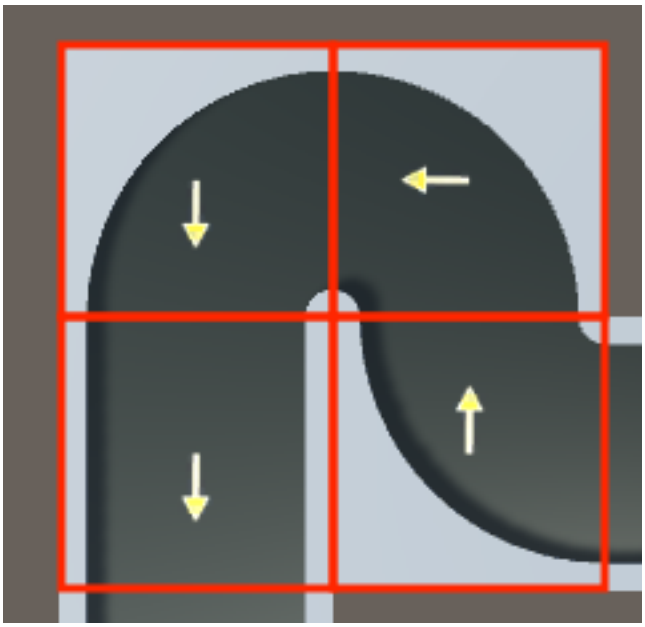

図 2 路面タイルの構成
Fig. 2 Composition of road surface tiles

表 1 ニューラルネットワークへの入力情報
Table 1 Inputs for neural network

自車情報	進行方向との角度 [rad]	[-1, 1]
	最大速度（v_{max}）	[0, ∞]
	最大ハンドル角（θ_{max}）	[0, ∞]
センサ情報	距離	[0, d_{max}]
	壁の有無	{0, 1}
	車の有無	{0, 1}
	横方向相対速度	$[-\infty, \infty]$
	縦方向相対速度	$[-\infty, \infty]$

された距離と，その物体が壁であるか（壁の場合は 1 を入力），車両であるか（車両の場合は 1 を入力）を 0/1 の 2 値で取得する．また，センサで検知した物体が車両の場合は，その車両の進行方向情報を取得し，自車と他車の相対速度を入力する．自車情報 3 入力，センサ情報 4 入力 ×8 センサの計 43 入力をニューラルネットワークの入力とする．

3.3 制御方法

本研究では，車両制御として，車両のアクセルとブレーキの制御による縦方向の制御と，ハンドル角の制御による横方向の制御を用いる．縦方向の制御は，最大速度を v_{max} とし，制御量 c_v $(-1 \leq c_v \leq 1)$ を調整することで縦方向移動量 $v_{max} \times c_v$ として制御する．横方向の制御は，最大ハンドル角を θ_{max} とし，制御量 c_θ $(-1 \leq c_\theta \leq 1)$ を調整することで横方向制御量 $\theta_{max} \times c_\theta$ として制御する．強化学習では，各状態に最適な制御量 c_v と c_θ を学習する．すべての車両は同じパラメータ θ を共有し，共通の重みを持つニューラルネットワークを用いて制御量を決定する．つまり，すべての車両は同じ方策に基づいて行動を決定する．

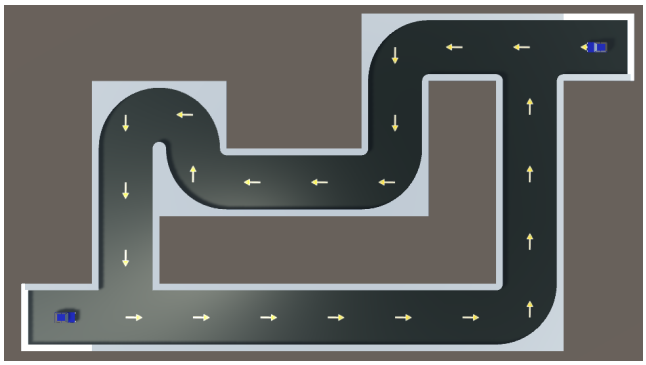

図 3 学習用コース
Fig. 3 Training course

3.4 報酬設計

本研究では，コース上に設置されたタイル間の移動に基づいて報酬を設計する．各車は隣接するタイルに移動したときに報酬として +1 を獲得する．また，同じ場所に停車することを避けるために，タイルの移動がない場合は −0.01 をペナルティとして与える．タイル間の移動に対する報酬に加え，各車はタイルの進行方向と自車の方向との誤差に応じて追加報酬を獲得する．追加報酬は，進行方向と自車方向のなす角度 θ を用いて $(1-\theta) \times \max(0, c_v)$ で算出する．ここで，c_v は縦方向の制御量であり，$c_v \leq 0$（後退）の場合は追加報酬は 0 とする．

一方，壁や他車との衝突を避けるために，ペナルティを与える．壁や他車と衝突した場合は，ペナルティとして −10 を獲得する．また，直前にいたタイルに戻る場合は −1 のペナルティをそれぞれ与える．衝突時のペナルティを移動時の報酬より大きく設定することで，衝突回避を優先して学習することを目指す．

以上をまとめると，報酬 r は以下の式で算出される：

$$r = r_f + r_\theta + r_p \tag{2}$$

$$r_f = \begin{cases} +1 & \text{if move to the next tile} \\ -0.01 & \text{otherwise} \end{cases} \tag{3}$$

$$r_\theta = (1-\theta) \times \max(0, c_v) \tag{4}$$

$$r_p = \begin{cases} -10 & \text{if crash} \\ -1 & \text{if move to the previous tile} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

4. 実験設定

図 3 に，学習時に用いるコースを示す．図中の矢印は車両の進行方向を示す制御マーカーであり，コースを反時計回りに周回する．学習中は左下タイルと右上タイルから一定時間間隔で新規車両がスタートする．車両は，他車，もしくは壁に衝突した場合に負の報酬を受け取り，初期地点（左下タイル，または右上タイル）から再スタートする．車両の最大速度 v_{max} は 5 から 15 の間でランダムに割り当

Vol.2021-DPS-187 No.14
Vol.2021-MBL-99 No.14
Vol.2021-ITS-85 No.14
2021/5/27

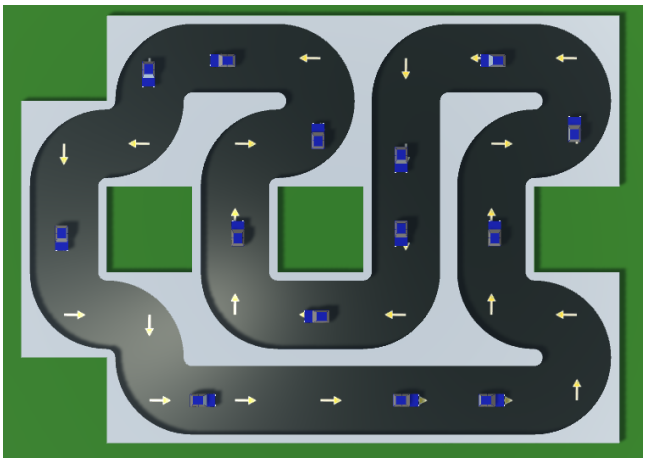

図 4 テスト用コース（14 台）
Fig. 4 Test course (14 vehicles)

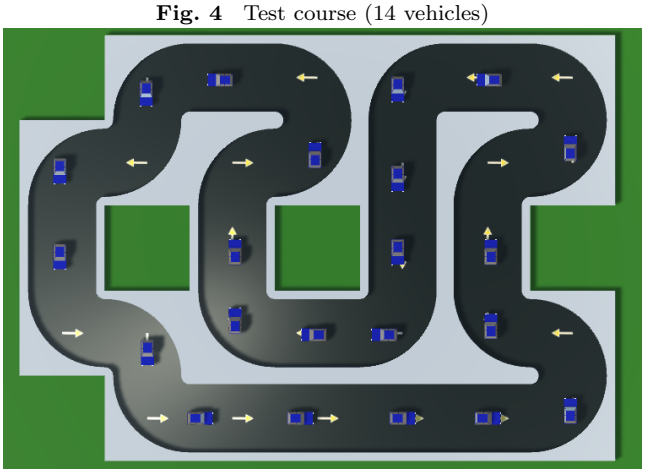

図 5 テスト用コース（21 台）
Fig. 5 Test course (21 vehicles)

てる．

図 4 と図 5 に，評価時に用いるコースを示す．評価時には，学習時とは異なる構造のコースを使用する．車両が 14 台走行する車両が疎なコース（図 4）と，車両が 21 台走行する車両が密なコース（図 5）を用いる．評価時は，新規車両の追加はない．車両が壁や他車と衝突した場合は初期位置から再スタートする．車両の最大速度 v_{max} は 5 から 15 の間でランダムに割り当てる．

PPO の実装には，Unity で利用可能な機械学習ツールキットである Unity Machine Learning Agents（ML-Agents）[3] を用いる．ML-Agents では，PPO を含む最新の強化学習アルゴリズムを利用可能であり，本研究ではこの実装を用いる．PPO の学習パラメータは表 2 に示すとおりである．また，実環境でのセンサ誤差を想定し，学習時と評価時にはセンサ値にそれぞれ ±10% の誤差を与える．

5. 結果

本章では，学習コースを用いた学習時の挙動と，学習の結果得られたパラメータを用いたテスト用コースでの挙動を分析する．

表 2 PPO の学習パラメータ
Table 2 Training parameters of PPO

パラメータ	値
バッチサイズ	1024
バッファサイズ	81920
学習率	3.0×10^{-4} (線形減衰)
β	1.0×10^{-4}
ϵ	0.2
λ	0.95
エポック数	3
隠れ層の数	2
隠れ層ニューロン数	256
学習ステップ数	3.0×10^{7}

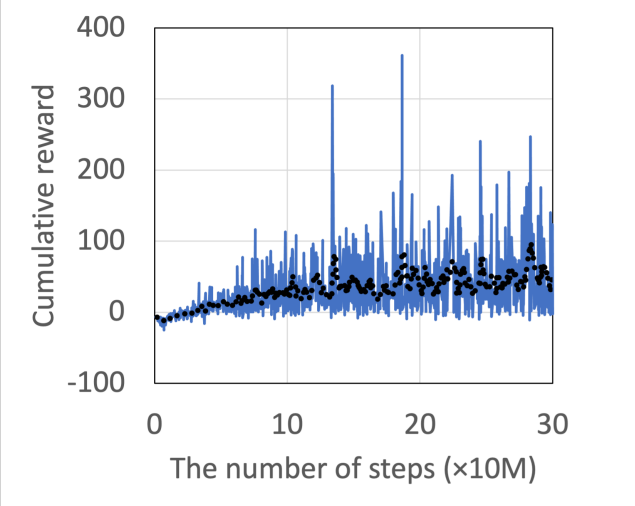

図 6 平均累積報酬
Fig. 6 Cumulative reward

5.1 学習過程

学習時の挙動を分析するために，学習時の平均累積報酬を図 6 に示す．図 6 において，横軸は学習ステップ数，縦軸は平均累積報酬を表す．累積報酬は 10^5 ステップごとに平均を取り，黒破線は 10 区間の移動平均を表す．結果から，学習初期は累積報酬が負になっており，衝突や走行の停滞によるペナルティを受けていることがわかる．一方，学習が進むにつれて累積報酬が増加しており，学習によってコースを衝突なく走行可能になっていることがわかる．このとき，学習用コースでは一定時間ごとに新しく車両が追加されることから，学習後期には複数車両がコース中を走行している場合でも正の報酬を得られていることがわかる．これらの結果から，PPO を用いた学習によって他車を考慮した走行制御が学習されているといえる．

5.2 テスト用コースでの挙動

PPO によって学習した車両制御の有効性を示すために，テスト用コースにおける平均周回時間と衝突率を計測する．

Vol.2021-DPS-187 No.14
Vol.2021-MBL-99 No.14
Vol.2021-ITS-85 No.14
2021/5/27

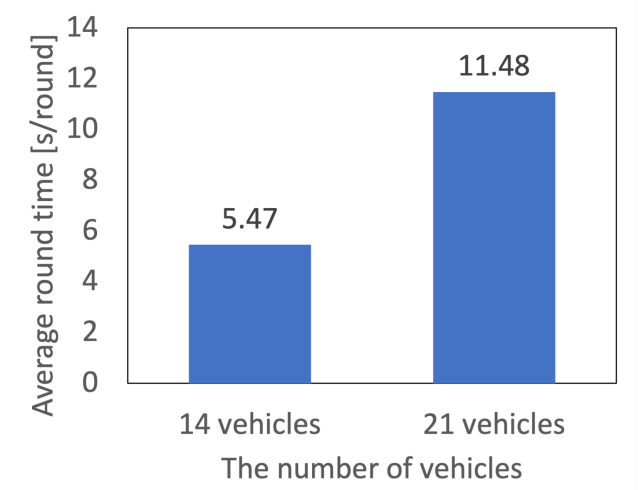

図 7 平均周回時間
Fig. 7 Average round time

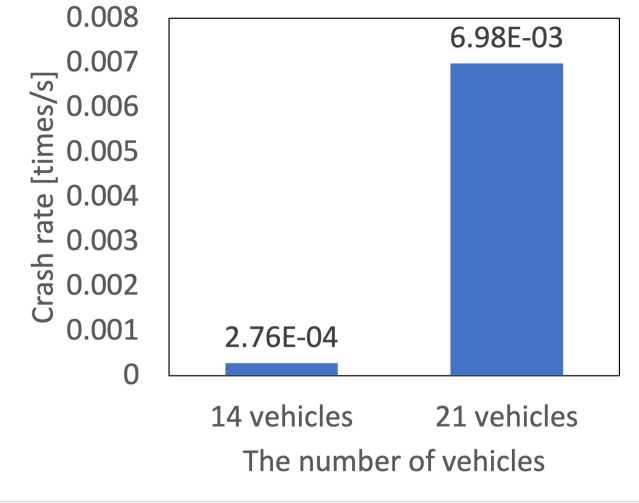

図 8 衝突率
Fig. 8 Crash rate

平均周回時間は，テスト用コースにおいて各車両がコースを 1 周するまでに要した平均シミュレーション時間を表し，式で算出する．

$$\hat{T} = \frac{T_{max} N_v}{n_t} \tag{6}$$

ここで，n_t はシミュレーション中にコース下部中央のタイルを通過した車両の台数，N_v はコース中の車両台数（図 4 では 14，図 5 では 21），T_{max} は総シミュレーション時間を表す．衝突率は，テスト用コースにおいて単位シミュレーション時間あたりに車両が壁，または他車に衝突した回数を表し，以下の式で算出する．

$$r_c = \frac{n_c}{N_v T_{max}} \tag{7}$$

ここで，n_c はシミュレーション中に車両が壁，または他車に衝突した回数を表す．

図 7 に平均周回時間を示す．横軸は，コース内の車両台数，縦軸は式 (6) で算出した平均周回時間を表す．シミュレーション実験の結果から，車両台数が 14 台の場合は 1 周 5.47s/周，21 台の場合は 1 周 11.48s/周 で周回できることがわかった．車両台数が少ない場合は，多い場合と比較してスムーズにコースを周回できている．特に，車両台数は 2/3 であるにも関わらず，周回時間は 1/2 以下になっていることがわかる．

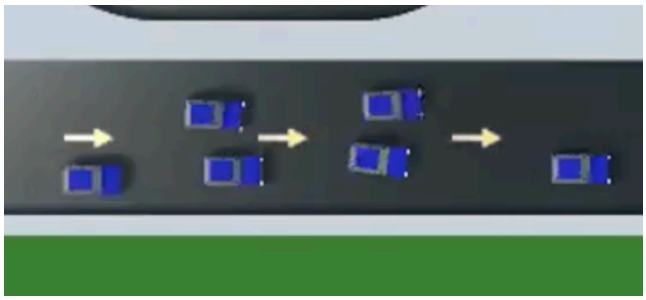

図 9 テスト用コースにおける追い越しの例
Fig. 9 An example of overtaking in the test course

図 8 に衝突率を示す．横軸は，コース内の車両台数，縦軸は式 (7) で算出した衝突率を表す．実験の結果から，車両台数が 14 台の場合は衝突率が非常に低く，衝突回数は 1 度だけであった．一方，車両台数が 21 台の場合は 14 台の場合と比べ，衝突率が 30 倍以上増加している．これは，コース中の車両密度が増加したことにより，特にコーナー付近での衝突が増加したためである．しかし，平均周回時間 11.48s/周 に対し，6.98×10^{-3}回/s であることから，車両の衝突が 1 回生じるまでの平均周回数は $1/(6.98 \times 10^{-3} \times 11.48) = 12.47$ 周/回 であり，約 12 周に 1 回程度の衝突回数である．このことから，車両が密な場合にも衝突率は低く抑えられているといえる．

6. 考察

21 台の車両が走行する図 5 のテスト用コースでは，図 9 に示すように，最大速度の速い車両が遅い車両の左側を追い抜く挙動が見られた．図中で，進行方向は右方向であり，右側（図中下部）を最大速度の遅い車両が走行し，その左側（図中上部）を最大速度の速い車両が追い越している．この挙動は，学習時に設計されたものではなく，複数車の協調学習によって創発したマクロ的な挙動である．

この挙動を詳細に分析するために，各車両の最大速度（v_{max}）と，他車が前方，または後方に検知された場合の横方向制御の制御量分布を図 10 と図 11 に示す．各図で横軸は横方向の制御量，縦軸は車両の最大速度（v_{max}）を表す．図 10 は車両の前方 3 つのセンサで他車が検知された場合の分布を表し，図 11 は車両の後方 3 つのセンサで他車が検知された場合の分布を表す．

図 10 から，速度の遅い車両は前方に他車が検知された場合は直進（制御量 0 付近）を高い割合で選択していることがわかる．一方，速度の速い車両は直進に加えて左方へ

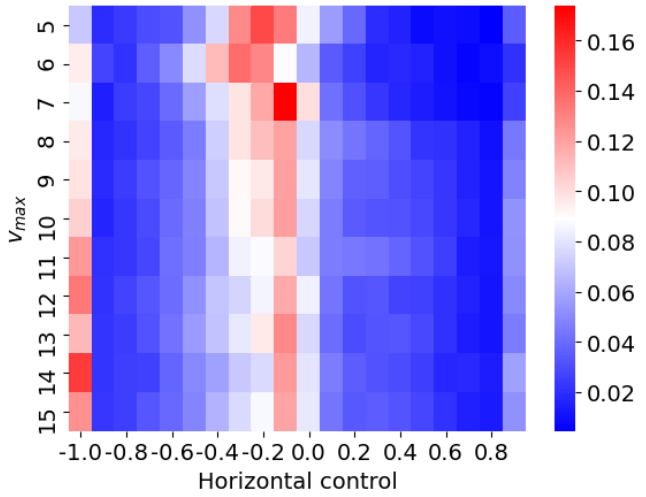

図 10 前方に車両を検知した場合の横方向制御の制御量分布
Fig. 10 Distribution of horizontal control when detecting other forward vehicles

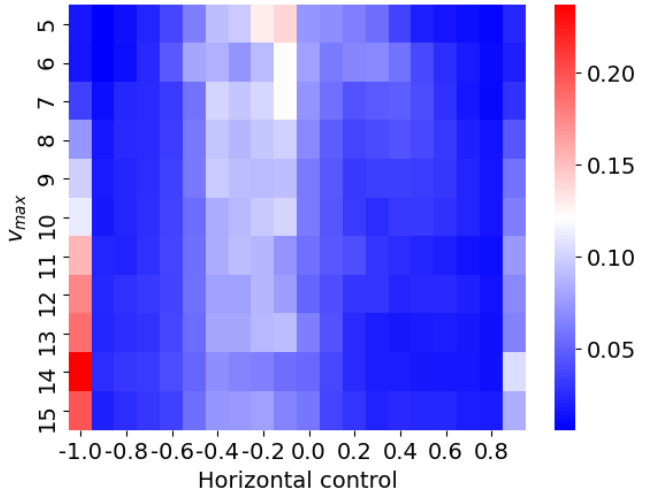

図 11 後方に車両を検知した場合の横方向制御の制御量分布
Fig. 11 Distribution of horizontal control when detecting other backward vehicles

の制御（制御量 −1 付近）を多く選択していることがわかる．一方，図 11 から，後方に他車を検知した場合も同様の傾向が見られる．速度の遅い車両は後方に車両を検知した場合は直進を高い割合で選択し，速い車両は左方への制御を多く選択していることがわかる．

以上の分析から，学習の結果，速度の速い車両が前後に他車が存在する場合に左方に移動することで衝突を回避しつつ，遅い車両は直進を維持する制御がなされている．これにより，図 9 に示すような追い越しが生じていると考えられる．

7. おわりに

本研究では，強化学習を用いて複数車が同一道路を走行する環境における自動運転制御の獲得に取り組んだ．特に，本研究では，距離センサから得られる情報と，近接車との相対速度の情報に基づく制御を提案した．

シミュレーション環境を用いて，深層強化学習アルゴリズムである PPO を用いた学習と評価を行った．実験の結果，提案アルゴリズムによって複数車が存在する環境における自動運転制御を獲得可能であることを示した．また，強化学習による自動運転制御の獲得によって，速度の速い車両が遅い車両の左側を追い抜くというマクロな協調行動が獲得されることを示した．

今後は，衝突率をより抑えつつスループットを向上させるための報酬設計の改良に取り組む．また，車載センサから得られる情報だけでなく，道路に設置された車両検知器から得られる情報を複合することで，より高精度な制御を実現する．さらに，今回の実験では簡易的なシミュレーション環境を用いたため，今後は物理シミュレーションを用いたより現実的な環境での実験に取り組むとともに，RC カーなどを用いた実機実験にも取り組む予定である．

参考文献

[1] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G. and Pineau, J.: An Introduction to Deep Reinforcement Learning, *Foundations and Trends® in Machine Learning*, Vol. 11, No. 3-4, pp. 219–354 (online), DOI: 10.1561/2200000071 (2018).
[2] Haas, J. K.: A history of the unity game engine (2014).
[3] Juliani, A., Berges, V.-P., Teng, E., Cohen, A., Harper, J., Elion, C., Goy, C., Gao, Y., Henry, H., Mattar, M. and Lange, D.: Unity: A General Platform for Intelligent Agents (2020).
[4] Pal, A., Philion, J., Liao, Y.-H. and Fidler, S.: Emergent Road Rules In Multi-Agent Driving Environments (2021).
[5] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O.: Proximal Policy Optimization Algorithms (2017).
[6] Streck, A.: Reinforcement Learning a Self-driving Car AI in Unity, https://towardsdatascience.com/reinforcement-learning-a-self-driving-car-ai-in-unity-60b0e7a10d9e (2021 年 4 月 8 日アクセス) (2020).
[7] Sutton, R. S. and Barto, A. G.: *Reinforcement learning: An introduction*, MIT press (2018).
[8] 小川一太郎, 横山想一郎, 山下倫央, 川村秀憲, 酒徳　哲, 柳原　正, 田中英明: Deep Q-Network による RC カー群の運動制御を実現する協調学習の提案, 人工知能学会全国大会論文集, Vol. JSAI2017, pp. 3I2OS13b5–3I2OS13b5（オンライン）, DOI: 10.11517/pjsai.JSAI2017.0_3I2OS13b5 (2017).
[9] 小川一太郎, 横山想一郎, 山下倫央, 川村秀憲, 酒徳　哲, 柳原　正, 大岸智彦, 田中英明: Deep Q-Network を用いた自動運転車のゆずりあいによる交通流の効率化, 人工知能学会全国大会論文集, Vol. JSAI2018, pp. 3Z204–3Z204（オンライン）, DOI: 10.11517/pjsai.JSAI2018.0_3Z204 (2018).
[10] 佐藤健哉, 橋本雅文, 菅沼直樹, 加藤真平, 芝　直之, 花井将臣, 高田広章, 天沼正行, 香名守道, 大石淳也: 協調型自動運転のための LDM グローバルコンセプト実証実験, 第 13 回 ITS シンポジウム (2015). 1-1B-10.