同時・複数の障害物回避における自動運転の獲得方策決定法

○津谷龍一　荒井幸代 （千葉大学）

Acquiring Policy to Avoid Multiple Moving Objects via Reinforcement Learning

∗R. Tsutani and S. Arai (Chiba University)

Abstract− One of the difficult tasks in realizing autonomous driving system is determining an appropriate action in the situations where a vehicle should avoid multiple obstacles simultaneously. In this study, we develop a system that realizes avoiding or following behavior without colliding with moving obstacles such as other vehicles and pedestrians. As an good example of these situation, we take a T-intersection where the system should be paid attention to on the left, right, and in front of the vehicle. Specifically, we consider the acquisition of avoiding or following behavior of by reinforcement learning considering the two objectives of speed and safety. The proposed method makes a vehicle to adjust the priority of obstacles by introducing a reward that considers the weighting of each objective.

Key Words: Autonomous driving, Traffic flow optimization, Reinforcement learning

1　はじめに

近年の「自動運転」技術は大きく進歩しており，自動駐車や走行時の車線逸脱システム，前方車両の追従などヒューマンエラーに起因する事故減少への貢献が期待されている．日本内閣府が採用するアメリカの SAE International の基準によれば，現在の自動運転技術は，「限定された条件のもとでシステムが全ての運転タスクを実施するが，緊急時などシステムから要請があれば運転者が操作を行う必要がある」[1] というレベル 3 に到達している．しかし，限られた条件下での自動運転のほとんどが追従や車線逸脱を防ぐものであり，市街地の環境で，他の自動車などの複数障害物を同時に考慮することを想定した衝突回避や追従行動の自動運転は実現されていない．

近年の自動運転に関する研究では，強化学習を用いた障害物回避を実現する試み [2] がみられる．強化学習は一つの目的に対して，その目的達成のための方策を試行錯誤を繰り返して学習する．特に深層学習と強化学習を組み合わせた深層強化学習は，深層学習による強力な関数近似と特徴抽出により状況に対する最適な行動の獲得が可能である．

本研究では，市街地の T 字路走行時を例として，左右前方にそれぞれ注意すべき他車，歩行者などの移動障害物が存在する場合に，回避や追従行動を実現するシステムを開発する．具体的には，スピードと安全性の二つの目的を考慮する多目的逐次意思決定問題として定式化することによって，対象物の回避あるいは，追従行動を深層強化学習によって獲得する．本研究では，各目的の重みづけを考慮した報酬を導入することによって障害物の優先度を調整し，状況に合わせた方策が獲得できることを検証する．

2　関連研究

自動車運転の実現に対しては，多くの研究や技術が提案されており，障害物については停止している物，および，移動する物，それぞれ扱っている．停車車両や車線などの固定障害物に対してはその陰にある見えない対象物の予想など，自動運転に向けて重要な要素であるが，実際の交通は，これに加えて対向車や前方車両など移動する障害物が多く存在し，運転者にはそれらとの衝突がない円滑な操作が求められる．

2.1　高速道路における自動運転

Yu ら [3] の研究では，片側 2 車線の高速道路の環境において，前方に走行する車両が 1 台存在する場合の車両を追い越す行動の学習に成功している．また，Zhu ら [4] の研究では片側 1 車線の高速道路の環境で，前方の人間が運転している自動車への追従行動の獲得に成功している．

2.2　市街地における自動運転

Jansonn ら [5] の研究では，片側 1 車線の市街地の環境で逆走車の回避行動や，前方を走行する自動車の追従行動の獲得に成功している．また，自動車会社 Tesla の車両に搭載されている「オートパイロット [6]」は，同一車線内でのハンドル操作，加速，ブレーキを自律的に行うシステムで，前方車両の追従が可能である．

2.3　同時に複数の障害物が存在する際の自動運転

Makantasis ら [7] の研究では，片側 3 車線の高速道路において同一方向を進行する車両が複数台存在する場合の追従・追い抜きの行動を獲得することに成功している．Tesla では「ナビゲート オン オートパイロット [8]」という技術が，オートパイロット [6] とは別に追加オプションとして車両に搭載できる．このシステムはカーナビゲーション設定時に，目的地までの高速道路でステアリング操作や加減速のみならず，車線変更を含めた自動運転を可能にしている．

2.4　本研究：市街地における同時・複数の障害物

移動障害物を考慮した既存研究の多くが一つの障害物に対するものであり，同時に複数の障害物を考慮した研究では，高速道路で同一方向に走行する障害物に対するもの [7] であった．一方，多くの事故は高速道路ではなく市街地で起こっており，その原因は対向車や歩行者，自転車などの混合交通流であることが挙げられる．混合交通流においては同時に複数の障害物を回避する場合が多く，これを考慮した自動運転が課題である．

そこで本研究では，市街地走行を想定し，対向車と横方向からの飛び出しが同時に発生する環境でそれらと衝突のない行動を獲得する手法を提案する．

3 強化学習

強化学習は，意思決定主体であるエージェントが，環境と相互作用しながら，幾度の試行錯誤を経て，行動によって与えられる報酬の累積値を最大になるように学習する手法である．

強化学習のモデル化は，マルコフ決定過程 (MDP) によって行われる．ある状態への遷移確率がその直前の状態のみに依存するというマルコフ性をもとにモデル化される．マルコフ決定過程は $\langle S, A, P, R, \gamma \rangle$ でモデル化される．$\mathcal{S}$ は環境の状態集合，$\mathcal{A}$ はエージェントの行動集合，$\mathcal{P}$ は遷移先の確率分布の集合，$\mathcal{R}$ は状態遷移後にエージェントが獲得する報酬の集合，そして γ は割引率を示す．

各離散時間 t において，エージェントは状態 $s_t \in \mathcal{S}$ を観測し，状態から選択可能な行動に対する選択確率分布である方策 $\pi(a_t|s_t)$ を用いて行動 $a_t \in A$ を選択する．エージェントは行動の実行後，環境から次状態 $s_{t+1} \in \mathcal{S}$ と報酬 $r(s_t, a_t) \in \mathcal{R}$ を受け取る．強化学習の目標は，初期状態から割引累積期待報酬を最大化する方策を学習することである．目的関数は式 (1) で表される．

$$J(\pi) = \mathbb{E}_{(s_t,a_t)\tilde{E},\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] \tag{1}$$

3.1 Proximal Policy Optimization (PPO)

強化学習は，Value-based と Policy-based に大別され，本研究では後者を用いる．Policy-based の手法は，価値関数を用いず，方策空間内で直接方策を最適化する．Policy-based の手法において代表的な方策勾配法では，方策 $\pi(a_t|s_t;\theta)$ を θ でパラメータ化し，θ の関数として目的関数 $J(\pi(a_t|s_t;\theta))$ を表現し，勾配降下法を用いて最適化する．方策勾配法では，勾配の推定値の分散が大きい場合，方策パラメータの更新が大きくなり，学習が安定しない場合がある．

方策勾配法に基づいた深層強化学習手法である Trust Region Policy Optimization(TRPO)[9] は，方策の更新幅を制限することによって上記の問題を回避する．TRPO は価値関数 V を学習し，方策勾配の計算に必要なアドバンテージ関数 A を推定する．学習中，TRPO は K ステップの間方策を実行し，行動価値関数 $Q(s,a)$，状態価値関数 $V(s)$ の推定値からアドバンテージ関数 $A(s,a)$ を式 (2) - (4) で計算する．

$$V_\pi(s_t) = \mathbb{E}_{a_t,s_{t+1},\ldots}\left[\sum_{l=0}^{K}\gamma^t r(s_{t+l})\right] \tag{2}$$

$$Q_\pi(s_t,a_t) = \mathbb{E}_{s_{t+1},a_{t+1},\ldots}\left[\sum_{l=0}^{K}\gamma^t r(s_{t+l})\right] \tag{3}$$

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s) \tag{4}$$

TRPO の目的関数では式 (5) のように，更新前の方策 $\pi_{\theta_{old}}$ と現在の方策 π_θ との間の Kullback-Leibler(KL) divergence の変化に応じてペナルティを与える．

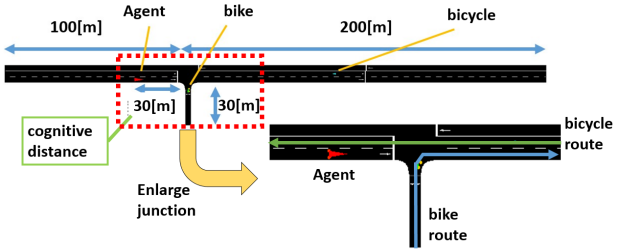

Fig. 1: junction environment

$$L^{TRPO}(\theta) = \hat{\mathbb{E}}_t[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t - \beta KL[\pi_{\theta_{old}}(\cdot\,|s_t), \pi_\theta(\cdot\,|s_t)]] \tag{5}$$

β はペナルティに対する重みである．TRPO はニューラルネットワークのパラメータが共有できず，実装が困難な手法として知られる．

Proximal Policy Optimization (PPO)[10] は KL divergence を用いる TRPO に対して実装が容易で拡張性のある近似手法である．PPO の目的関数は式 (6) で表される．

$$L^{PPO}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{6}$$

ここで，$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ は方策の変化量であり，$r_t(\theta)$ をパラメータ ϵ によって一定の範囲内にクリッピングすることで，方策の大きな変化を防ぐ．本手法では，強化学習手法として PPO を用いる．

4 提案手法

4.1 実験環境

実験ではシミュレータとして，環境作成が容易なことやパラメータが充実しているという観点から，SUMO[11] を用いる．実験は，Fig. 1 に示す片側 1 車線で制限速度が 60[km/h] の道路において行った．1 ステップを 0.1[s] として実験し，障害物は，エージェントとなる自動車の進行方向に対して右側前方の側道から飛び出しエージェントと同一進行方向を走行するバイクと，対向車線をエージェントに向かう方向に走行する自転車の二つである．

実験で用いた環境はエージェントの走行車線 2 車線，対向車線 1 車線の構造になっている．これはシミュレータ上でエージェントを逆走させることができなかったためで，対向車線を走行する自転車を逆走させることによって，見かけ上走行車線 2 車線で片側 1 車線の道路環境を表現した．

この環境においてエージェントは実験環境の左端から速度 10[m/s] で走行を開始し，バイクは側道の下端から速度 3[m/s] で出現しエージェントが交差点前約 25[m] 付近に到達した時に側道からエージェントの走行車線に飛び出す．エージェントはバイクの位置を交差点前 30[m] になるまで観測できず，対向車線を走行する自転車の位置は常に観測できるものとする．また，バイクと自転車の最高速度はそれぞれ 8[m/s]，4.17[m/s] に設定し，バイクは最高速度に達するまで 0.8[m/s^2] で加速する (自転車は出現時から最高速)．

4.2 状態入力・行動出力・報酬設計

状態入力を Table 1，行動出力を Table 2 に示す．報酬関数は Table 3 に示すように設定した．バイク，自転車の位置と速度をそれぞれ $x_{bike}, y_{bike}, v_{bike}$，$x_{bicy}, y_{bicy}, v_{bicy}$ とする．

状態入力は，エージェントとなる車が観測可能な状態として，エージェント自身の道路環境での位置，速度，加速度に加え，二つの障害物との相対位置，距離，相対速度の計 12 次元を用いた．障害物との相対位置，相対速度はエージェントの位置，速度から障害物の位置，速度を引いたものを用いる．距離の計算式を式 (7)，(8) に示す．

行動出力は，エージェントの車が実行できる行動として，前ステップの状態を維持する「何もしない」，0.8[m/s^2] での「加速」，5[m/s^2] での「減速」，「車線変更」の 4 種類とした．また，車線変更には 1[s](10 ステップ) かかるものとし，その際 0.3[m/s^2] 減速するものとした．

報酬設計はスピード罰の R_1，対向車線走行罰の R_2，障害物と衝突した時の罰である R の 3 種類とした．R_1 はエージェントの速度 v_{car} から道路の制限速度 (16.7[m/s]) を減算し 50 で割ったものを用いる．R_2 は対向車線を走行した時にステップごとに-0.8 を与える．

$$dist_{bike} = \sqrt{(x_{car} - x_{bike})^2 + (y_{car} - y_{bike})^2} \quad (7)$$

$$dist_{bicy} = \sqrt{(x_{car} - x_{bicy})^2 + (y_{car} - y_{bicy})^2} \quad (8)$$

Table 1: state

State input
car position(x_{car}, y_{car})
car speed (v_{car})
car acceleration (a_{car})
Relative position of car and bike(x_{sbike}, y_{sbike})
Relative position of car and bicycle(x_{sbicy}, y_{sbicy})
Distance of car and bike $(dist_{bike})$
Distance of car and bicycle $(dist_{bicy})$
Relative velocity of car and bike (v_{sbike})
Relative velocity of car and bicycle (v_{sbicy})

Table 2: action

Action output
do-nothing
accel(+0.8[m/s^2])
brake(-5[m/s^2])
lane-change(-0.3[m/s^2])

Table 3: reward

	setting	value
R_1	speed penalty	$(v_{car} - 16.7)/50$
R_2	driving opposite lane penalty	-0.8
R	collision penalty	-100

4.3 多目的逐次意思決定問題としての定式化

本研究では複数の障害物が同時に存在する際の運転方策獲得問題を，Table 3 に示す報酬関数 R_1, R_2 による多目的逐次意思決定問題として定式化する．エージェントは各目的に対する重み (W_1, W_2) を持つ．この二つの重みを変更することで，異なる方策を獲得する．報酬の合計 R_{all} を，式 (9) に示す．

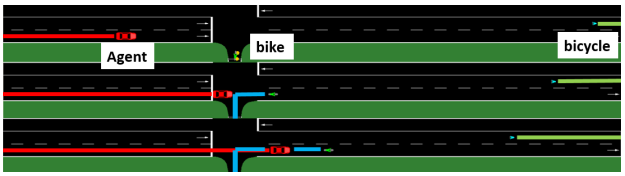

Fig. 2: following

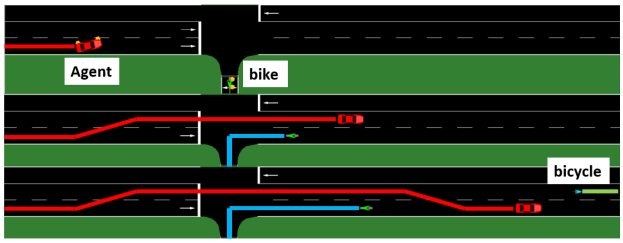

Fig. 3: overtaking

$$\begin{aligned} R_{all} &= R_1 W_1 + R_2 W_2 + R \\ W_1 + W_2 &= 1 \\ 0 \le W_1 &\le 1 \\ 0 \le W_2 &\le 1 \end{aligned} \quad (9)$$

5 計算機実験

Fig. 1 に示す環境において，11 通りの重みの組み合わせに対し，それぞれ 300,000 ステップ学習させた．

5.1 実験結果と考察

実験結果を Table 4 に示す．また，追従と追い抜きについてシミュレーション画面の様子をそれぞれ Fig. 2, Fig. 3 に示す．

軌跡図は赤色がエージェント軌跡，青色がバイクの軌跡，緑色が自転車の軌跡を示しており，縦軸は位置の y 座標 [m] を，横軸は位置の x 座標 [m] を示す．軌跡図の下半分は位置ごとに選択した行動を表しており，下から順に「何もしない」(do-nothing),「加速」(accel),「減速」(brake),「車線変更」(lane-change) である．また，速度推移図は，赤色がエージェントの速度を示しており，青の点線はバイクの最高速度 8[m/s] を示す．横軸は時間 [s] を，縦軸は速度 [m/s] を示す．

Table 4: result

W_1	W_2	action result
0.0	1.0	following bike$(v_{sbike} < 0)$
:	:	:
0.4	0.6	following bike(keep following distance)
0.5	0.5	overtaking bike・avoidance bicycle
:	:	:
0.9	0.1	overtaking bike・avoidance bicycle
1.0	0.0	meandering after avoidance obstacles

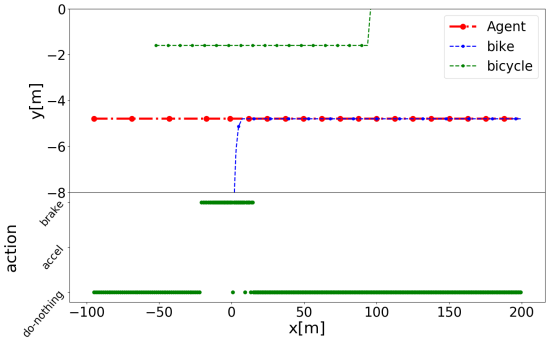

Fig. 4: following trajectory(slow)

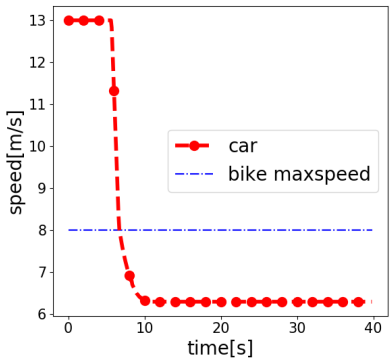

Fig. 5: following speed(slow)

重み W_1 が 0.0 から 0.3 の時は，エージェントはバイクを認知後ブレーキをかけ，十分に減速してから追従行動をとる方策を獲得した．W_1 が 0.1 の時の軌跡を Fig. 4 に，エージェントの車両速度を Fig. 5 に示す．エージェントの速度はバイクの最高速 (青線) に比べ小さく，自転車の追従速度も遅い．これは対向車線に出た際の罰が大きく，スピードによる罰も小さいためであると考えられる．また，バイクより遅い速度での追従になったのは，ステップ当たりの罰が $W_1 = 0.1$ の場合-0.02 程度であり，衝突罰-100 に比べ極めて小さいため，衝突を確実に避けるためであると考えられる．

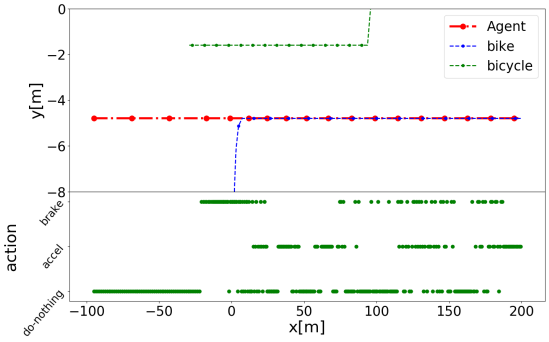

Fig. 6: following trajectory

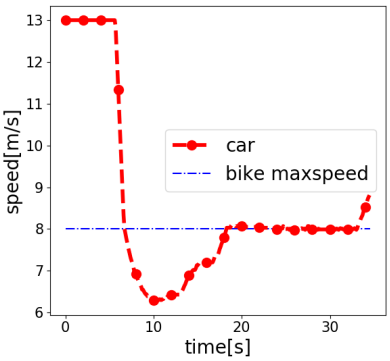

Fig. 7: following speed

W_1 が 0.4 の時の軌跡を Fig. 6 に，エージェントの車両速度を Fig. 7 に示す．エージェントは追従行動を獲得したが，バイク認知後ブレーキをかけ，バイクの後を等車間距離で追従する方策を獲得した．これは重み W_1 が 0.0 から 0.3 の場合に比べスピードの罰が大きいため，速度を落とさないように行動したからと考えられる．

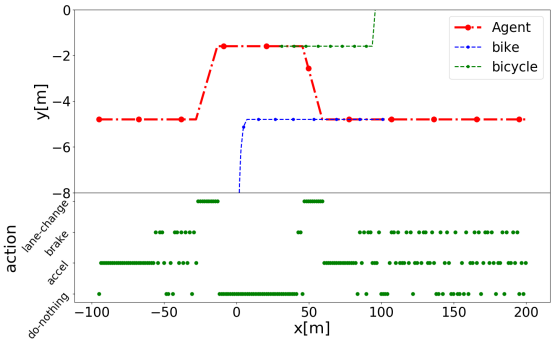

Fig. 8: overtaking trajectory

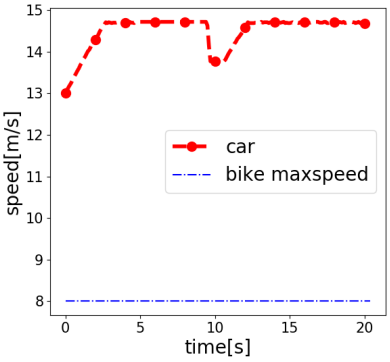

Fig. 9: overtaking speed

重み W_1 が 0.5 から 0.9 の時は，バイク追い抜き，自転車を回避する方策を獲得した．W_1 が 0.6 の時の軌跡を Fig. 8 に，エージェントの車両速度を Fig. 9 に示す．バイク認知後に車線変更し，反対車線に飛び出して加速した上で，バイクを追い抜いた後自転車と衝突する前に走行車線に戻る行動を取った．エージェントの速

度はバイクの最高速に比べ大きく，バイクを追い抜き反対車線に出た後，自転車の前でブレーキをかけて走行車線に戻るような回避行動を取っていることが分かる (W_1 が 0.9 に近付くほど減速は少ない)．これはスピードの罰が大きくなり，減速することの罰が大きくなるため，なるべく速度を落とさずに行動するために反対車線に飛び出したと考えられる．

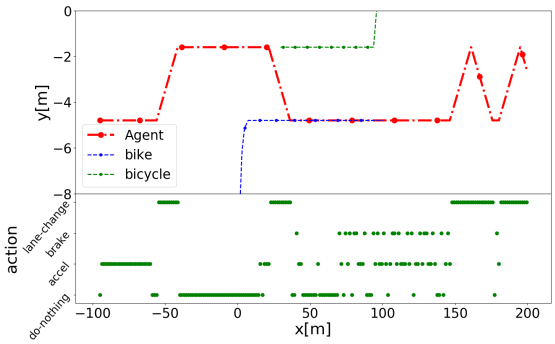

Fig. 10: meandering trajectory

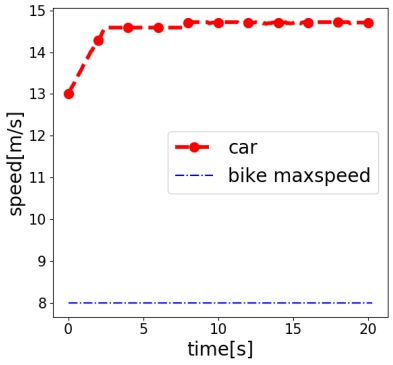

Fig. 11: meandering speed

W_1 が 1.0 の時の軌跡を Fig. 10 に，エージェントの車両速度を Fig. 11 に示す．重み W_1 が 1.0 の時には，W_1 が 0.5 から 0.9 と同様にバイクと自転車を追い抜いた後，蛇行運転をした．速度推移は開始時点から高い速度を維持したまま終了していることが分かる．これは反対車線の走行罰がないためと考えられる．

6 まとめ

本研究では，二つの障害物を同時に回避する問題を，スピードと危険度の 2 目的による多目的逐次意思決定問題として定式化し，目的の重みの違いによって異なる獲得方策が得られることを示した．この結果をふまえて，市街地でも周辺の住民の特性，たとえば高齢者が多い地域や通学路などに合わせて，重みを調整したシステム実装が考えられる．

今後の課題として，蛇行行動などの妥当でない行動の防止策のための報酬関数の改善などが挙げられる．また，横方向の加速度を制限し，乗り心地も考慮した運転行動の獲得に向けた応用などが考えられる．

参考文献

1) SAE International. Taxonomy and definitions foe tearms related to driving automation systems for on-road motor vehicles(sae j3016:sep2016). In *SURFACE VEHICLE RECOMMEND PRACTICE*, 2016/09.

2) Hyunmin Chae, Chang Mook Kang, ByeoungDo Kim, Jaekyum Kim, Chung Choo Chung, and Jun Won Choi. Autonomous braking system via deep reinforcement learning. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6. IEEE, 2017.

3) Lingli Yu, Xuanya Shao, Yadong Wei, and Kaijun Zhou. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning. *Sensors*, Vol. 18, No. 9, p. 2905, 2018.

4) Wen-Xing Zhu and HM Zhang. Analysis of mixed traffic flow with human-driving and autonomous cars based on car-following model. *Physica A: Statistical Mechanics and its Applications*, Vol. 496, pp. 274–285, 2018.

5) Jonas Jansson, Jonas Johansson, and Fredrik Gustafsson. Decision making for collision avoidance systems. *SAE Transactions*, pp. 197–204, 2002.

6) Tesla. オートパイロット. In *https://www.tesla.com/jp/autopilot*, 2021.

7) Konstantinos Makantasis, Maria Kontorinaki, and Ioannis Nikolos. A deep reinforcement learning driving policy for autonomous road vehicles. *arXiv preprint arXiv:1905.09046*, 2019.

8) Tesla. ナビゲートオンオートパイロット. In *https://www.tesla.com/jp/blog/introducing-more-seamless-navigate-autopilot*, 2019.04/03.

9) John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

10) John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

11) Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie WieBner. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582. IEEE, 2018.