
Freeway Merging in Congested Traffic based on Multipolicy Decision Making with Passive Actor Critic

Tomoki Nishi^{*1} Prashant Doshi² Danil Prokhorov³

Abstract

Freeway merging in congested traffic is a significant challenge toward fully automated driving. Merging vehicles need to decide not only how to merge into a spot, but also where to merge. We present a method for the freeway merging based on multi-policy decision making with a reinforcement learning method called *passive actor-critic* (pAC), which learns with less knowledge of the system and without active exploration. The method selects a merging spot candidate by using the state value learned with pAC. We evaluate our method using real traffic data. Our experiments show that pAC achieves 92% success rate to merge into a freeway, which is comparable to human decision making.

1. Introduction

Highly automated vehicles are growing in popularity (Goo; Ziegler et al., 2014; Aeberhard et al., 2015; Okumura et al., 2016) after DARPA2007 Urban Challenge (DARPA, 2007). Merging in congested traffic is one of the tricky challenges toward fully automated driving. The difficulty is caused by three issues: how to model ambient vehicles, how to decide a merging spot and how to merge into the spot.

Multipolicy decision making (MPDM) has been developed in (Galceran et al., 2017). The approach decides based on multiple candidates with the scores of each policy according to user-defined cost function by using the forward simulation. Modeling ambient vehicles in congested traffic is needed for the forward simulation which is very difficult especially in congested traffic.

Reinforcement learning is a powerful framework to learn policy without prior knowledge of environment by using

active exploration. The score in MPDM can be calculated without forward simulation by using cumulative expected reward called state value as the score. The reinforcement learning, however, needs to explore the environment to optimize policy. An exhaustive exploration of the state and action spaces is impossible because it would be unacceptable for an autonomous vehicle to try maneuvers which lead to a crash or even a near-miss, leaving unexplored much of the state and action spaces.

Nishi et al. (2017) developed a reinforcement learning method called *passive Actor Critic* (pAC), to optimize policy using data collected under passive dynamics and knowing dynamic model of ego-vehicle instead of such explorations. They showed that the method could learn improved policies in the highway merging domain with simulation and real traffic dataset. However in their experiment, the merging spot was decided in advance.

We develop a method for the freeway merging maneuver in congested traffic based on multi-policy decision making with pAC. The policy to merge into each candidate of merging spots is learned with pAC from collected data and ego-vehicle dynamics model. The method can choose a policy without forward simulation using instead the state-value function estimated by pAC.

2. Preliminaries

We focus on a discrete-time system with a real-valued state $\mathbf{x} \in \mathbf{R}^n$ and control input $\mathbf{u} \in \mathbf{R}^m$, whose stochastic dynamics is defined as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + A(\mathbf{x}_k)\Delta t + B\mathbf{u}_k\Delta t + \text{diag}(\boldsymbol{\sigma})\boldsymbol{\Delta}\omega, \quad (1)$$

where $\boldsymbol{\Delta}\omega$ is differential Brownian motion simulated by a Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}\Delta t)$, where \mathbf{I} is the identity matrix. $A(\mathbf{x}_k)$, $B\mathbf{u}_k$ and $\boldsymbol{\sigma} \in \mathbf{R}^n$ denote the *passive* dynamics, *control* dynamics due to action, and the transition noise level, respectively (B is an input-gain matrix). Δt is a step size of time and k denotes a time index. System dynamics structured in this way are quite general: for example, models of many mechanical systems conform to these dynamics.

L-MDP (Todorov, 2009a) is a subclass of MDPs defined by a tuple, $\langle \mathcal{X}, \mathcal{U}, \mathcal{P}, \mathcal{R} \rangle$, where $\mathcal{X} \subseteq \mathbf{R}^n$ and $\mathcal{U} \subseteq \mathbf{R}^m$ are

¹Toyota Central R & D Labs., Inc., Nagakute, Aichi, Japan
²THINC Lab, Dept. of Computer Science, University of Georgia, Athens, GA 30622 ³Toyota R & D, Ann Arbor, MI 48105.
Correspondence to: Tomoki Nishi <nishi@mosk.tytlabs.co.jp>.

continuous state and action spaces. $\mathcal{P} := \{p(\mathbf{y}|\mathbf{x}, \mathbf{u}) | \mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}\}$ is a state transition model due to action, which is structured as in Eq. 1, and $\mathcal{R} := \{r(\mathbf{x}, \mathbf{u}) | \mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}\}$ is an immediate cost function with respect to state \mathbf{x} and action \mathbf{u} . A control policy $\mathbf{u} = \pi(\mathbf{x})$ is a function that maps a state \mathbf{x} to an action \mathbf{u} . The goal is to find a policy that minimizes the following average expected cost: $V_{avg} := \lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E} \left[\sum_{k=0}^{n-1} r(\mathbf{x}_k, \pi(\mathbf{x}_k)) \right]$.

Grondman et al. (2012) notes that the Bellman equation for MDPs can be rewritten using the value function $V(\mathbf{x})$ called *V-value*, state-action value function $Q(\mathbf{x}, \mathbf{u})$ called *Q-value*, and average value V_{avg} under an policy.

$$V_{avg} + Q_k = r_k + \mathbb{E}_{p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k)}[V_{k+1}]. \quad (2)$$

As we may expect, $V_k = \min_{\mathbf{u} \in \mathcal{U}} Q_k$. $\mathbb{E}_{p(\mathbf{x}_{k+1}|\mathbf{x}_k)}[\cdot]$ denotes expectation over a probability distribution of state transition under the passive dynamics. Here and elsewhere, subscript k denotes values at time step k .

An L-MDP defines the cost of an action (control cost) to be the amount of additive stochasticity:

$$r(\mathbf{x}_k, \mathbf{u}_k) := q(\mathbf{x}_k)\Delta t + KL(p(\mathbf{x}_{k+1}|\mathbf{x}_k)||p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k)). \quad (3)$$

Here, $q(\mathbf{x}) \geq 0$ is the state-cost function; $KL(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence; $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ models the *passive* dynamics while $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k)$ represents the *active* or control dynamics of the system. L-MDPs further add a condition on the dynamics as shown below.

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k) = 0 \Rightarrow \forall \mathbf{u}_k p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k) = 0.$$

This condition ensures that no action introduces new transitions that are not achievable under passive dynamics. In other words, actions are seen as simply contributing to the passive dynamics. The stochastic dynamical system represented by Eq. 1 satisfies this assumption naturally because the dynamic is Gaussian. However, systems that are deterministic under passive dynamics remain so under active dynamics. This condition is easily met in robotic systems.

The standard Bellman equation for MDPs can then be recast in L-MDPs to be a linearized differential equation for exponentially transformed value function of Eq. 2 (hereafter referred to as the linearized Bellman equation) (Todorov, 2009b):

$$Z_{avg} Z_k = e^{-q_k \Delta t} \mathbb{E}_{p(\mathbf{x}_{k+1}|\mathbf{x}_k)}[Z_{k+1}], \quad (4)$$

where $Z_k := e^{-V_k}$ and $Z_{avg} := e^{-V_{avg}}$. Here, Z_k and Z_{avg} are an exponentially transformed value function called *Z-value* and the average cost under an optimal policy, respectively.

Because the passive and control dynamics with the Brownian noise are Gaussian, the KL divergence between these dynamics becomes

$$KL(p(\mathbf{x}_{k+1}|\mathbf{x}_k)||p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k)) = 0.5 \mathbf{u}_k^\top S^{-1} \mathbf{u}_k \Delta t, \quad (5)$$

where $S^{-1} := B^\top (\text{diag}(\boldsymbol{\sigma}) \text{diag}(\boldsymbol{\sigma})^\top)^{-1} B$. Then, the optimal control policy for L-MDPs can be derived as,

$$\pi(\mathbf{x}_k) = -SB^\top \frac{\partial V_k}{\partial \mathbf{x}_k}. \quad (6)$$

3. Multi-policy decision making with pAC

We present a novel MPDM algorithm with pAC. MPDM determines control input by selecting a policy in multiple candidates with the scores of each policy. While prior MPDM algorithm requires forward simulation to score each policy candidates, our algorithm scores the candidates without the simulation by using instead state value estimated with pAC. We provide details on the policy selection and pAC below.

3.1. MPDM policy selection with pAC

The MPDM policy selection returns the best policy based on a set of policy candidates and observed current state. Algorithm 1 shows the policy selection algorithm. The algorithm sets Π of available policies. Score c of each candidate, which is calculated using state value estimated with pAC, is added to the set of scores C . Finally, the optimal policy associated with the minimum score is returned as the best policy.

Algorithm 1 MPDM policy selection with pAC

```

Set  $\Pi$  of available policies.
 $C \leftarrow \emptyset$ 
for  $\pi_i \in \Pi$  do
    Set current state:  $\mathbf{x}_k \leftarrow x$ .
    Calculate score of a policy  $\pi_i$  learned by pAC:  $c_i \leftarrow \hat{V}_i(\mathbf{x}_k)$ 
     $C \leftarrow C \cup c_i$ 
end for
Choose the best policy  $\pi^* \in \Pi$ :  $\pi^* \leftarrow \arg \min_{\pi_i \in \Pi} C$ 
return  $\pi^*$ 
    
```

3.2. Passive Actor-Critic for L-MDP

We introduce an actor-critic method for continuous L-MDP, which we label as *passive actor-critic* (pAC). While the actor-critic method usually operates using samples collected actively in the environment (Konda & Tsitsiklis, 1999), pAC finds a converged policy without exploration. Instead, it uses samples of passive state transitions and a known control dynamics model. pAC follows the usual two-step schema of actor-critic: a state evaluation step (critic), and a policy improvement step (actor):

1. **Critic:** Estimate the Z-value and the average cost from the linearized Bellman equation using samples under passive dynamics;
2. **Actor:** Improve a control policy by optimizing the Bellman equation given the known control dynamics model, and the Z-value and cost from the critic.

We provide details about these two steps below.

Estimation by Critic using Linearized Bellman

The critic step of pAC estimates Z-value and the average cost by minimizing the least-square error between the true Z-value and estimated one denoted by \hat{Z} .

$$\begin{aligned} \min_{\nu, Z_{avg}} \quad & \frac{1}{2} \int_{\mathbf{x}} \left(\hat{Z}_{avg} \hat{Z}(\mathbf{x}; \nu) - Z_{avg} Z(\mathbf{x}) \right)^2 d\mathbf{x}, \\ \text{s.t.} \quad & \int_{\mathbf{x}} \hat{Z}(\mathbf{x}; \nu) d\mathbf{x} = C, \forall \mathbf{x} \quad 0 < \hat{Z}(\mathbf{x}; \nu) \leq \frac{1}{Z_{avg}}, \end{aligned} \quad (7)$$

where ν is a parameter vector of the approximation and C is a constant value used to avoid convergence to the trivial solution $\hat{Z}(\mathbf{x}; \nu) = 0$ for all \mathbf{x} . The second constraint comes from $\forall \mathbf{x}, Z(\mathbf{x}) := e^{-V(\mathbf{x})} > 0$ and $\forall \mathbf{x}, q(\mathbf{x}) \geq 0$. The latter implies that $V + V_{avg} > 0$, with $Z_{avg} Z(x) := e^{-(V+V_{avg})}$ which is less than 1.

We minimize the least-square error in Eq. 7, $\hat{Z}_{avg} \hat{Z}_k - Z_{avg} Z_k$, with TD-learning. The latter minimizes TD error instead of the least-square error that requires the true $Z(\mathbf{x})$ and Z_{avg} , which are not available. The TD error denoted as e_k^i for linearized Bellman equation is defined using a sample $(\mathbf{x}_k, \mathbf{x}_{k+1})$ of passive dynamics as, $e_k^i := \hat{Z}_{avg}^i \hat{Z}_k^i - e^{-q_k} \hat{Z}_{k+1}^i$, where the superscript i denotes the iteration. \hat{Z}_{avg} is updated using the gradient as follows:

$$\hat{Z}_{avg}^{i+1} = \hat{Z}_{avg}^i - \alpha_1^i \frac{\partial (e_k^i)^2}{\partial \hat{Z}_{avg}^i} = \hat{Z}_{avg}^i - 2\alpha_1^i e_k^i \hat{Z}_k^i, \quad (8)$$

where α_1^i is the learning rate, which may adapt with iterations.

In this work, we approximate the Z-value function using a neural network (NN). The parameters ν are updated with the following gradient based on backpropagation:¹

$$\frac{\partial}{\partial \nu^i} \left(\hat{Z}_{avg} \hat{Z}_k^i - Z_{avg} Z_k \right)^2 \approx 2e_k^i \hat{Z}_{avg}^i \frac{\partial \hat{Z}_k^i}{\partial \nu^i}, \quad (9)$$

where e_k^i is the TD error as defined previously.

Actor Improvement using Standard Bellman

¹ $e^{-\tanh(x)}$ or $e^{-\text{softplus}(x)}$ is used as an activation function of the output layer to satisfy the constraint $\hat{Z} \geq 0$. The constraint $\int_{\mathbf{x}} \hat{Z}(\mathbf{x}; \nu) d\mathbf{x} = C$ is ignored in practice because convergence to $\forall \mathbf{x}, \hat{Z}(\mathbf{x}; \nu) = 0$ is rare. $\min([1, e^{-q_k} \hat{Z}_{k+1}^i])$ is used instead of $e^{-q_k} \hat{Z}_{k+1}^i$ to satisfy $\hat{Z} \leq 1/Z_{avg}$ in Eq. 7.

The actor improves a policy by computing S (Eq. 5) using the estimated Z-values from the critic because we do not assume knowledge of noise level σ . It is estimated by minimizing the least-square error between the V-value and the state-action Q-value:

$$\min_S \frac{1}{2} \int_{\mathbf{x}} \left(\hat{Q}(\mathbf{x}, \hat{\mathbf{u}}(\mathbf{x})) - V(\mathbf{x}) \right)^2 d\mathbf{x},$$

where V is the true V-value and \hat{Q} is the estimated Q-value under the estimated action $\hat{\mathbf{u}}(\mathbf{x})$. Notice from Eq. 6 that a value for S results in a policy as B is known. Thus, we seek the S that yields the optimal policy by minimizing the least-square error because the Q-value equals V-value iff $\hat{\mathbf{u}}$ is maximizing.

Analogously to the critic, we minimize the least-square error given above, $\hat{Q}_k^i - V_k$, with TD-learning. To formulate the TD error for the standard Bellman update, let \mathbf{x}_{k+1} be a sample at the next time step given state \mathbf{x}_k under passive dynamics, and let $\tilde{\mathbf{x}}_{k+1} := \mathbf{x}_{k+1} + B\hat{\mathbf{u}}_k \Delta t$ be the next state using control dynamics. Rearranging terms of the Bellman update given in Eq. 2, the TD error d_k^i becomes

$$d_k^i := r(\mathbf{x}_k, \hat{\mathbf{u}}_k) + \hat{V}^i(\tilde{\mathbf{x}}_{k+1}) - \hat{V}_{avg} - \hat{V}_k^i.$$

We may use Eqs. 3 and 5 to replace the reward function,

$$\begin{aligned} r(\mathbf{x}_k, \hat{\mathbf{u}}_k) &= (q_k + 0.5 \hat{\mathbf{u}}_k^\top (\hat{S}^i)^{-1} \hat{\mathbf{u}}_k) \Delta t, \\ &= q_k \Delta t + 0.5 \frac{\partial \hat{V}_k^i}{\partial \mathbf{x}_k}^\top B \hat{S}^i B^\top \frac{\partial \hat{V}_k^i}{\partial \mathbf{x}_k} \Delta t. \end{aligned}$$

The last step is obtained by noting that $\hat{\mathbf{u}}_k^i = -\hat{S}^i B^\top \frac{\partial \hat{V}_k^i}{\partial \mathbf{x}_k}$, where \hat{S} denotes estimated S in Eq. 6.

The estimated V-value and its derivative is calculated by utilizing the approximate Z-value function from the critic. \hat{S} is updated based on standard stochastic gradient descent using the TD error,

$$\hat{S}^{i+1} = \hat{S}^i - \beta^i \frac{\partial}{\partial \hat{S}^i} \left(\hat{Q}_k^i - V_k \right)^2 \approx \hat{S}^i - 2\beta^i d_k^i \frac{\partial d_k^i}{\partial \hat{S}^i},$$

where β is the learning rate. The actor mitigates the impact of error from estimated Z-value by minimizing the approximated least-square error between V- and Q-values under the learned policy.

Algorithm

We show a pseudo code of pAC in Algorithm 2. $Z(\mathbf{x})$ and Z_{avg} are estimated in the critic with samples, and S is done in the critic with samples, estimated $\hat{Z}(\mathbf{x})$ and \hat{Z}_{avg} . In the critic, feedback from the actor is not needed (unlike actor critic methods for MDPs) because the Z-value is approximated with samples from passive dynamics only. We

Algorithm 2 passive Actor Critic

Initialize parameters $\hat{Z}_{avg}^0, \nu^0, \hat{S}^0, \alpha_1^0, \beta^0$
for Iteration $i = 1$ to N **do**
 Sample set of a state, a next state and state cost $(\mathbf{x}_k, \mathbf{x}_{k+1}, q_k)$
 from dataset randomly
 critic:
 $e_k^i \leftarrow \hat{Z}_{avg}^i \hat{Z}_k^i - \exp(-q_k) \hat{Z}_{k+1}^i$
 Update ν^{i+1} with $2e_k^i \hat{Z}_{avg}^i \frac{\partial \hat{Z}_k^i}{\partial \nu^i}$
 $\hat{Z}_{avg}^{i+1} \leftarrow \hat{Z}_{avg}^i - 2\alpha_1^i e_k^i \hat{Z}_k^i$
 actor:
 $\hat{V}_k^i \leftarrow -\ln \hat{Z}_k^i, \hat{V}_{k+1}^i \leftarrow -\ln \hat{Z}_{k+1}^i, \hat{V}_{avg}^i \leftarrow -\ln \hat{Z}_{avg}^i$
 $d_k^i \leftarrow q_k + 0.5 \frac{\partial \hat{V}_k^i}{\partial \mathbf{x}_k} B \hat{S}^i B^\top \frac{\partial \hat{V}_k^i}{\partial \mathbf{x}_k} \Delta t + \hat{V}_{k+1}^i - \hat{V}_{avg}^i - \hat{V}_k^i$
 $\hat{S}^{i+1} \leftarrow \hat{S}^i - 2\beta^i d_k^i \frac{\partial d_k^i}{\partial \hat{S}^i}$
end for

emphasize that the actor and critic steps do not use the functions A and σ but does indeed rely on B , of Eq. 1. As such, the updates use a sample $(\mathbf{x}_k, \mathbf{x}_{k+1})$ of the passive dynamics, and the state cost q_k .

4. Freeway merging based on MPDM

We present a freeway merging algorithm based on MPDM with pAC. The algorithm learns a policy and a state value function to merge into a predetermined spot with pAC from collected dataset in advance. The algorithm determines a merging spot from a set of candidates and control input with the learned model when an autonomous vehicle is driving on a merging ramp. We provide details on learning the policy and merging based on the learned policy below.

4.1. Learning a merging policy in a predetermined spot

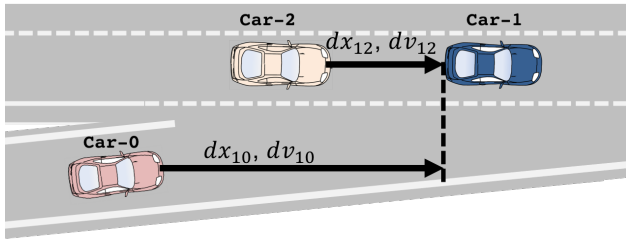


Figure 1. The 3-car system for merging. Merging vehicle (Car-0) should ideally merge midway between the following vehicle (Car-2) and leading vehicle (Car-1). dx_{10} and dv_{10} denote Car-0's relative position and velocity from Car-1.

The algorithm learns a policy in a predetermined spot based on pAC with the following state space, action space and state cost function. We refer the reader to Fig. 1 for our notation in the four-dimensional state space. Here, $\mathbf{x} = [dx_{12}, dv_{12}, dx_{10}, dv_{10}]^\top$ where dx_{ij} and dv_{ij} denote the horizontal signed distance and relative velocity between cars i and $j \in [0, 1, 2]$. The action space is one-dimensional

(acceleration). The state cost is designed to motivate Car-0 to merge midway between Car-1 and Car-2 with the same velocity as Car-2:

$$q(\mathbf{x}) = k_1 - k_1 \exp \left(-k_2 \left(1 - \frac{2dx_{10}}{dx_{10}} \right)^2 - k_3 dv_{12}^2 \right),$$

$$k_1 = 1, k_2 = 10, k_3 = 10 \text{ if } dx_{12} < dx_{10} < 0,$$

$$k_1 = 10, k_2 = 10, k_3 = 0 \text{ otherwise,}$$

where k_1, k_2 and k_3 are weights for the state cost.

4.2. Freeway merging based on the learned policy

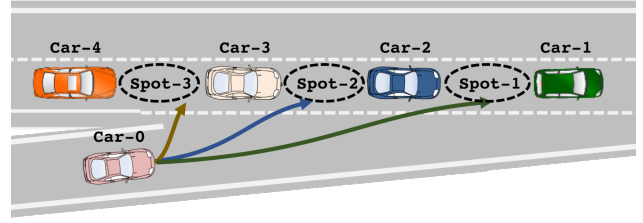


Figure 2. A typical freeway merging situation. There are three mergeable spot candidates: Spot-1, 2 and 3. The merging vehicle needs to determine a spot from a set of the candidates and control input to merge.

We present how to merge onto a freeway based on the learned policy. Figure 2 shows a typical freeway merging situation. In this situation, there are three possible mergeable spots: Spot-1, 2 and 3. First, the algorithm finds three mergeable spot candidates: Spot-1, 2 and 3. Three 3-car systems associated with the each spot are then extracted: $\{\text{Car-0, Car-1, Car-2}\}$, $\{\text{Car-0, Car-2, Car-3}\}$ and $\{\text{Car-0, Car-3, Car-4}\}$.

The best policy is selected with Algorithm 1. The state value function learned to merge into a predetermined spot can be used to calculate the scores of any spot candidates because the MDP is the same and only states are different between these candidates.

5. Experiment on real-world traffic

We evaluate our algorithm with NGSIM data set recorded real-world traffic on freeway.

5.1. Problem settings

The NGSIM data set contains vehicle trajectory data recorded by cameras mounted on top of a building on eastbound Interstate-80 in the San Francisco Bay area in Emeryville, CA for 45 minutes. It is segmented into three 15-minute periods around the evening rush hour (NGSIM-dataset, 2005). In these periods, one can observe a transition from non-congested to moderately congested to full congestion. Vehicle trajectories were extracted using a vehicle



Figure 3. A snapshot of vehicles tracked in NGSIM. The bounding boxes denote the positions of tracked vehicles (their numbers are NGSIM indices).

tracking method from collected videos (Kovvali et al., 2007). We extracted 5-car system (Fig. 2) representing 637 freeway merging events, when human drivers always merged into Spot-2, and applied a Kalman smoother to mitigate vehicle tracking noise. We calculated next states \mathbf{x}_{k+1} under passive dynamics by subtracting state change caused by actions: $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^D - B^\top \mathbf{u}_k^D$, where \mathbf{x}_{k+1}^D and \mathbf{u}_k^D are the next state and the action recorded in the data set respectively.

5.2. Results

Figure 4 shows an example of freeway merging behavior with our algorithm. The merging vehicle tried to merge behind Car-2 at $T=0$ seconds. The vehicle then switched to merge behind Car-1 and succeeded merging into the spot. The switching would be reasonable because the vehicle would have needed hard braking if the vehicle had merged behind Car-2 due to large velocity gap between Car-0 and Car-2.

We compared a policy based on the proposed method to three policy to merge into each of the predetermined spots: Spot-1, 2 and 3. The left chart of Fig. 5 shows average costs of each policy to merge into Spot-1, 2 and 3. Our method achieved comparable average cost to the policy to merge into Spot-2 chosen by human drivers, and outperform other policies. The right chart of Fig. 5 shows a success rate of freeway merging into each spot. The result also shows that our method can select proper merging spots because the performance is better than that of the policy to merge into Spot-2. Our method fails occasionally when a following vehicle strongly accelerates to reduce the gap or a leading vehicle strongly decelerates.

6. Concluding Remarks

We presented a novel MPDM method with pAC. The method needs no forward simulation because it uses estimated state value with pAC unlike prior MPDM methods. This feature would be preferable when we install our MPDM method onto autonomous vehicles because smaller computational resources would be needed to calculate the score with estimated state value function than for any approach employing forward simulations.

We also illustrated our MPDM on a freeway merging task. The algorithm determines a merging spot and a policy to merge into a predetermined spot with the learned model and a state value function after extracting all possible 3-car systems.

We evaluated the method using freeway merging domain on real traffic data. Our method achieved a 92% success rate, which is comparable to merging success if the spot is selected by human drivers. We are interested in improving performance of our method further, as well as exploring additional challenges, e.g., deciding when to enter roundabouts.

References

- Google self-driving car project. <https://www.google.com/selfdrivingcar/>.
- Aeberhard, Michael, Rauch, Sebastian, Bahram, Mohammad, Tanzmeister, Georg, Thomas, Julian, Pilat, Yves, Homm, Florian, Huber, Werner, and Kaempchen, Nico. Experience, results and lessons learned from automated driving on germanys highways. *IEEE ITS Magazine*, 7 (2):42–57, 2015.
- DARPA. Darpa urban challenge. <http://archive.darpa.mil/grandchallenge/>, 2007.
- Galceran, Enric, Cunningham, Alexander G, Eustice, Ryan M, and Olson, Edwin. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 2017. ISSN 0929-5593. doi: 10.1007/s10514-017-9619-z.
- Grondman, Ivo, Busoniu, Lucian, Lopes, Gabriel AD, and Babuska, Robert. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, 2012.
- Konda, Vijay R and Tsitsiklis, John N. Actor-critic algorithms. In *Advances in neural information processing systems 13*, pp. 1008–1014, 1999.
- Kovvali, Vijay Gopal, Alexiadis, Vassili, Zhang, PE, et al. Video-based vehicle trajectory data collection. In *Transportation Research Board 86th Annual Meeting*, 2007.
- NGSIM-dataset. <http://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm>, 2005. *U.S Department of Transportation, Federal Highway Administration*, Accessed August 20, 2016.
- Nishi, Tomoki, Doshi, Prashant, James, Michael, and Danil, Prokhorov. Actor critic for linearly-solvable con-

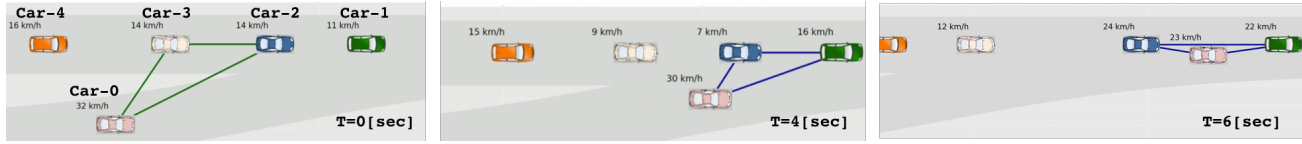


Figure 4. An example of freeway merging behavior with our algorithm. Triangles denote the 3-car system with the merging vehicle. **(Left)** The vehicle tried to merge behind Car-2. **(Center)** The vehicle switched to merge behind Car-1 because it is easier to merge into the spot than others according to the scores calculated with MPDM. The switching would be reasonable because the vehicle would have needed hard braking if the vehicle had merged behind Car-2 due to large velocity gap between Car-0 and 2. **(Right)** The vehicle is succeeding to merge behind Car-1.

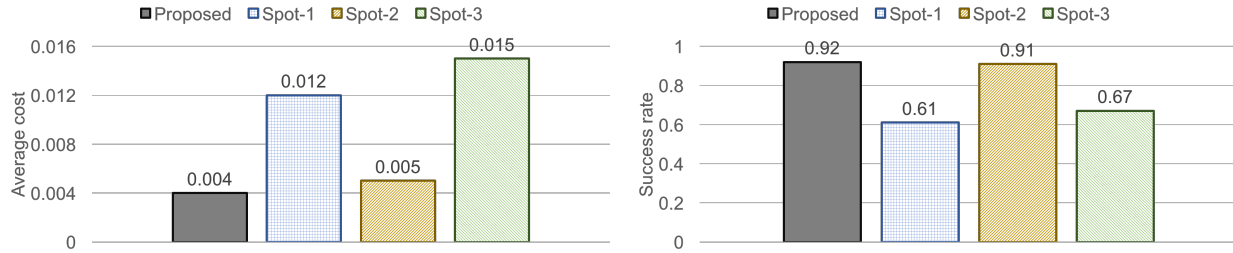


Figure 5. **(Left)** Average costs of each policy. The policy with the proposed method(gray) is comparable to a policy to merge into Spot-2 (orange) chosen by human drivers. **(Right)** Success rate of each policy. Our method achieved a 92% success rate. The performance is comparable to a success rate of the policy to merge always into Spot-2.

tinuous mdp with partially known dynamics. *arXiv preprint:1706.01077*, 2017.

Okumura, Bunyo, James, Michael R, Kanzawa, Yusuke, Derry, Matthew, Sakai, Katsuhiro, Nishi, Tomoki, and Prokhorov, Danil. Challenges in perception and decision making for intelligent automotive vehicles: A case study. *IEEE Transactions on Intelligent Vehicles*, 1(1):20–32, 2016.

Todorov, Emanuel. Efficient computation of optimal actions. *Proceedings of the national academy of sciences*, 106(28): 11478–11483, 2009a.

Todorov, Emanuel. Eigenfunction approximation methods for linearly-solvable optimal control problems. In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pp. 161–168. IEEE, 2009b.

Ziegler, Julius, Bender, Philipp, Schreiber, Markus, Lategahn, Henning, Strauss, Tobias, Stiller, Christoph, Dang, Thao, Franke, Uwe, Appenrodt, Nils, Keller, C, et al. Making bertha drive: An autonomous journey on a historic route. *Intelligent Transportation Systems Magazine, IEEE*, 6(2):8–20, 2014.