Accepted By IV 2017, Draft Version

Intention Estimation For Ramp Merging Control In Autonomous Driving

Chiyu Dong[1], John M. Dolan[2], and Bakhtiar Litkouhi[3]

***Abstract*— Cooperative driving behavior is essential for driving in traffic, especially for ramp merging, lane changing or navigating intersections. Autonomous vehicles should also manage these situations by behaving cooperatively and naturally. In this paper, we present a novel learning-based method to efficiently estimate other vehicles' intentions and interact with them in ramp merging scenarios, without over-the-air communication between vehicles. The intention estimate is generated from a Probabilistic Graphical Model (PGM) which organizes historical data and latent intentions and determines predictions. Real driving trajectories are used to learn transition models in the PGM. Thus, besides the structure of the PGM, our method does not require human-designed reward or cost functions. We validate the performance of our method in both real merging data and designed merging strategy in simulation, which show significant improvements compared with previous methods. The new method is computationally efficient, and does not require acceleration information about other vehicles, which is hard to read directly from sensors mounted on the autonomous vehicle.**

I. INTRODUCTION

Since the DARPA Urban Challenge, there has been significant work on developing autonomous urban driving. Some of this work is currently being commercialized. Early advanced driving assistance systems (ADAS) can detect dangers and warn drivers. For example, lane-departure warning can advise the driver if the car drifts too much from the lane center; collision detection can warn the driver if the car is too close to other cars. The most advanced products can assume control in specific simple scenarios. For example, GM's Full Speed Range ACC and Audi's "STOP and GO" adaptive cruise control (ACC) enable the car to follow other cars even in dense traffic at low speed. Mercedes-Benz's lane departure prevention system and Tesla's "Autopilot" combine ACC with auto-steering to achieve a certain level of autonomous driving at high speed.

Though those techniques can allow cars to drive hands-free under certain conditions, they do not guarantee proper interaction with other cars. Even if autonomous cars become affordable to consumers and successfully commercialized in the near future, there will be a long period of time before human-driven cars disappear. It is therefore important for autonomous cars to exhibit social behaviors to properly interact with human-driven cars or other autonomous cars

*This work was supported by GM funding

[1]Chiyu Dong is with Department of Electrical and Computer Engineering Carnegie Mellon University, Pittsburgh, PA 15213 USA

[2]John M. Dolan is with the The Robotics Institute ,Carnegie Mellon University, Pittsburgh, PA 15213 USA

[3]Bakhtiar Litkouhi is with GM R&D Center, Warren, MI 48090 USA

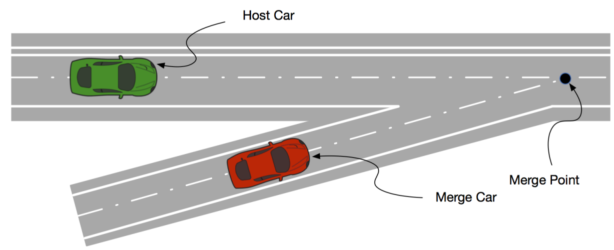


Fig. 1: Merge scenario. The host car is an autonomous vehicle, running on the main road; the merge car is a human-driven car, running on the ramp.

which are not equipped with a V2V communication system. The autonomous car should handle various cooperative situations, such as lane changing, intersections and entrance ramps. The biggest challenge of social behavior systems is to estimate human drivers' intentions. Human-driven cars introduce great uncertainty in autonomous-human driving interactions.

One typical example of these interactions is ramp merging. Normally, a human driver will implicitly "negotiate" with one or more drivers on a ramp, estimate their intentions, and then make decisions to successfully cooperate with them to pass the ramp merging point. Autonomous cars should make a decision to yield or not yield to the merging car which is based on some information. In this paper, we focus on ramp merge control, as shown in Fig. 1. The green car denotes the host vehicle (autonomous car) on the main road and the red car denotes a merging car (the human-driven car) on the ramp. The goal of our method is to estimate whether or not the merging car intends to yield to the host car, and then react to it.

II. RELATED WORK

There are several references that address the merging problem. Urmson et al. [1], Hidas [2] and Marinescu et al. [3] all used the same idea of a slot-based approach for cooperative merging control. They first check merging availability for each slot in the target-lane (a slot is the free area between two cars). Then they check feasibility of actions to find the best feasible slot for acceptable merging acceleration. Their decision is based on current states and no historical data are considered, which can lead to failures in some cases. Details will be discussed in Section III-B.2.

J. Wei et al. [4] proposed an intention-integrated framework to enable an autonomous car to perform cooperative social behavior. Accelerations of cars merging from a ramp are considered. The estimation again only considers the merging vehicle's current state, ignoring its historical state. The lack of historical data leads to instability in estimated intention, which results in oscillation or delayed reaction to the autonomous vehicle. To react to surrounding vehicles and reduce computational time, Wei et al. [5] proposed a QMDP [6] single-lane behavior framework which takes uncertainties into account. They also applied a cost function to evaluate and select the proper strategy. The Markov Decision Process (MDP) implicitly estimates intention based only on current state, again without considering historical data. Alexander et al. [7] extended the reaction ability of autonomous cars from a single lane to multiple lanes, including merging. Their multipolicy decision-making method a partially observable Markov decision process (POMDP) where no historical data are considered. Moreover, its probabilities, reward function and cost-functions require manual design. Schlechtriemen et al. [8] proposed a learning-based approach to estimate lane-changing intention. They calculate lane-changing probabilities by vehicles' lateral speeds using Random-Decision-Forest and Gaussian Mixture Regression. Whereas we predict merging intention by vehicle's longitudinal speeds over time using graphical model. Lenz et al. [9] generate cooperative planning for autonomous highway driving using Monte-Carlo Tree Search (MCTS) which relies on a manually designed cooperative cost function. Their method can handle multiple vehicle interactions in a merging scenario in the simulator. However, all vehicles in the simulator behave upon the designed cost function and model. Lenz did not validate the method in massive real world data.

Prior work has focused on current state and neglected historical data. One possible reason is that involving more data dramatically increases the dimension of the parameters, which makes the computation intractable. This paper improves on prior work by combining historical data, current state, and movement prediction. To decrease the dimension of the parameters, a probabilistic graphical model (PGM) is applied to describe dependency among the data. The PGM can clearly organize relationships among historical data, movement prediction and intention estimation. Thus the joint distribution among the data can be separated into several conditionally independent distributions, which eases analysis and computation. Besides the structure of the PGM, our method does not require other human-designed parameters or cost functions. We rely on real driving data to parametrize this model. The intention estimation can be used in various cooperation situations, such as lane changing, stop sign traversal and ramp merging control.

III. PGM-BASED INTENTION ESTIMATION

A. Structure of PGM

Intuitively, human drivers estimate intentions of other cars by their current and immediately previous state and

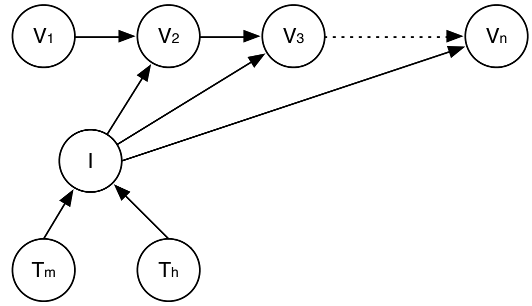


Fig. 2: Probabilistic Graphical Model of the social behavior of an autonomous vehicle.; V_n is the current speed, V_i is the speed at the previous time step; T_m, T_h are the current time-to-arrival for merging and host car respectively; I is the latent intention which needs to be estimated.

by considering the driving environment, such as road topology. Our method simulates this process to achieve human-like social behavior. The most important part of our method is understanding the cause-effect relationship among previous states and intention. To simplify and abstract this dependency, we apply a probabilistic graphical model. There are three kinds of nodes in the model: (1) state nodes, which are the time-to-arrival for each car; (2) an intention node, which is either "Yield" or "Not Yield"; (3) speed nodes, which contain the speed history of the target vehicle. The model's topology describes the dependency. Intuitively, current state affects intention, thus speed changes. So we design the graphical model as shown in Fig. 2. The task here is to estimate the intention node once the car has observed enough information (speeds and times-to-arrival). As the program runs, the speed nodes are updated by the speeds of the last n cycles and the time-to-arrival nodes are updated by the current speeds and the distance-to-merging-point for each car.

Our model assumes that human intention does not oscillate as fast as the program's update rate. Therefore, one intention node will affect the next n speed nodes. These n speed nodes keep track of the target vehicle's speed during n cycles. The time-to-arrival for each car will initially decide the intention. However, this decision is solely based on current states. To further adjust the intention estimate, more evidence is needed. The speeds in the last n cycles can provide movement information of the car, and thus refine the intention estimate. Physically, given intention, those speeds form a Markov Chain, which means that every speed node is affected only by its parent node (the vehicle's previous speed).

B. Evaluation of PGM

We are interested in the following probability of the merging car's yielding or not yielding to the autonomous car.

$$P(I|\mathbf{V}, T_m, T_h) \tag{1}$$

In Equation 1, $\mathbf{V}$ denotes a vector of speed during n cycles:

$$\mathbf{V} = [V_1, V_2, ..., V_n] \tag{2}$$

The T_m, T_h are the time-to-arrival for the merging and host cars, respectively. I denotes the estimated intention of the target vehicle.

$$\begin{aligned} P(I|\mathbf{V}, T_m, T_h) =& \frac{P(I, \mathbf{V}, T_m, T_h)}{P(\mathbf{V}, T_m, T_h)} \\ &\propto P(\mathbf{V}, T_m, T_h|I)P(I) \end{aligned} \tag{3}$$

Equation 3 is based on Bayes' Rule, and we focus on the likelihood term. From the graphical model, it is known that $\mathbf{V}$ and T_m, T_h are conditionally independent, given intention. Therefore this term can be further separated into two parts:

$$P(\mathbf{V}, T_m, T_h|I) = P(\mathbf{V}|I)P(T_m, T_h|I) \tag{4}$$

The first term is the speed term, and the second one is the time term. Besides these two terms, we also rely on the prior information $P(I)$ in Equation 3.

1) Speed term: From the graphical model, the merging speed has the Markov Property given intention. Thus the speed term can be further simplified:

$$\begin{aligned} P(\mathbf{V}|I) &= P(V_1, V_2, ..., V_n|I) \\ &= P(V_1|I)P(V_2|V_1, I)...P(V_n|V_{n-1}, I) \end{aligned} \tag{5}$$

Here we assume V_1, I are independent, thus $P(V_1|I) = P(V_1)$. Since there is no preference for V_1, it can be assumed to have a uniform distribution, namely $P(V_1) = \alpha$. To prevent underflow, log-likelihood is used:

$$\log P(\mathbf{V}|I) = \alpha \sum_{i=2}^{n} \log P(V_i|V_{i-1}, I) \tag{6}$$

There are only two intentions to be considered: yield and not yield, i.e., $P(\mathbf{V}|I = Y)$ and $P(\mathbf{V}|I = N)$. The probability distribution $\mathbf{P}$ will be learned directly from training data, which will be introduced in Section IV.

2) Time term: The time term implicitly contains two kinds of information: 1) current speed; 2) distance to the merge point. The time term describes how soon the cars will reach the merging point given current speed and distance. Values in the time term determine the intention of the merging vehicle. For example, if a shorter time is estimated for the host vehicle to reach the merging point, the merging vehicle will tend to yield to the host vehicle and this intention will result in a speed decrease of the merging vehicle. The time term will also reduce ambiguity in the speed term. In iPCB [4], only instantaneous acceleration is considered. Fig. 3 shows a failure scenario for iPCB that results from solely considering the current acceleration. This is a non-trivial problem which contributes the majority of the failure cases in the iPCB algorithm, as shown in Table II. However, in our proposed model, we additionally consider the time term, avoiding the ambiguity described above. The time term $P(T_m, T_h|I)$ is a

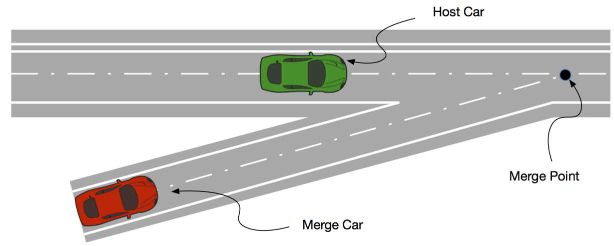


Fig. 3: Merging vehicle following behind but accelerating. The host (green) vehicle is on the main road; the merging vehicle (red) is on the on-ramp. In fact, the host vehicle can slightly accelerate to avoid ambiguities and collisions. However, the iPCB model will slow down the host vehicle regardless of the distance from the merging vehicle to the host and the current speed. The PGM model sends proper commands by integrating speed and distance information.

joint conditional probability, where T_m, T_h denote the time-to-arrival of the merging and host car, respectively. Time-to-arrival is defined as the current distance to the merging point divided by the current speed. To prevent underflow, we instead use $\log P(T_m, T_h|I)$.

3) Prior Term: In Equation 3, the last term $P(I)$ is the prior distribution for the merging vehicle intention, i.e., the percentage of merging vehicles that yield ($P(I = Y) = \gamma$) or not yield ($P(I = N) = 1 - \gamma$). This prior term gives an initial statistical estimate of intentions.

C. Intention Estimation Procedure

The final step is to combine the speed term and time term. Equations 3 and 4 yield:

$$\begin{aligned} \log P(I|\mathbf{V}, T_m, T_h) &= \log P(\mathbf{V}, T_m, T_h|I)P(I) \\ &= \log P(\mathbf{V}|I)P(T_m, T_h|I)P(I) \\ &= \underbrace{\alpha \sum_{i=2}^{n} \log P(V_i|V_{i-1}, I) +}_{\textit{Speed Term}} \\ &\quad \underbrace{\log P(T_m, T_h|I)}_{\textit{Time Term}} + \\ &\quad \underbrace{\log P(I)}_{\textit{Prior Term}} \end{aligned} \tag{7}$$

The estimated intention is:

$$I^* = \arg\max_{I} \ \log P(I|\mathbf{V}, T_m, T_h) \tag{8}$$

I^* is either "Yield" or "Not Yield". If "Yield", the program will send an acceleration command to an LQR controller [10]. Otherwise, a deceleration command is sent.

Algorithm 1 shows the full procedure for the intention estimation using the proposed PGM. We use a fixed-length queue Q to keep track of historical speed data. A new speed will be pushed to the end of the queue Q. If the size of Q is greater than length n, then the first element will be popped. This data structure ensures the program observes at most the last n cycles. The required state information S includes: current speeds and distances to the merging points of both

Algorithm 1 PGM-based intention estimation.

1: **procedure** INITIALIZATION(n)
2: $Q \leftarrow$ Queue with fixed-sized (n).
3: **procedure** EVALUATE PGM
4: **while** New Status S Comes in **do**
5: Push new speed to Q.
6: $P(Y) \leftarrow calculatePGM_Y(Q, S)$.
7: $P(N) \leftarrow calculatePGM_N(Q, S)$.
8: **if** $P(Y) > P(N)$ **then**
9: Command $\leftarrow$ Controller(Accelerate,S)
10: **else**
11: Command $\leftarrow$ Controller(Decelerate,S)

cars. The current speeds and distances are used to calculate the time term in Equation 7. Lines 6 and 7 calculate the probability of the Yield $P(Y)$ or Not Yield $P(N)$ intention of the other car, by using the method introduced in Section III-B. If $P(Y) > P(N)$, the host vehicle accelerates and tries to reach the merging point earlier than the merging car.

IV. TRAINING FROM DATA

In [4], prediction of the merging car's behavior was based on hand-coded cost functions and assumptions about the probability distribution of acceleration. We instead use the US-101 freeway real-world dataset NGSIM [11] to extract a model of cooperative behavior between host and merging vehicles. The dataset was obtained from overview cameras near the US-101 Ventura Boulevard entrance ramp in the Los Angeles area, as shown in Fig. 4. Cars in this region were filmed and tracked during morning rush hours (7:50 am to 8:35 am). The road segment consists of 5 lanes and one entrance ramp at the beginning. Vehicles in the 5th lane are considered host vehicles, and counterparts on the entrance ramp are considered merging vehicles. We preprocessed the data to filter out unrelated cars that run in inner lanes without interacting with merging vehicles, and used only those from the right-most lane and the entrance ramp. Host vehicles are paired with merging vehicles that are close to and temporally overlapped with the host. There were 354 host-merge vehicle pairs in the dataset. We use 1/3 of the total dataset for training, and the remaining 2/3 for real-data testing. We classify merging vehicles into two classes: 1) yield; 2) not yield, based on which car reaches the merging point first. From group 1, the distribution of $P(V|I = Y)$ can be estimated; from group 2, the distribution of $P(V|I = N)$.

A. Speed Transition Model

The goal of training is to estimate the conditional probability of intentions given historical speed information, i.e., $P(V|I)$. The two classes of data are used to train two different models, i.e., speed transition probability distributions. Fig. 5 shows examples of speed transition probability distributions for given speeds and intentions. The vertical dashed line indicates the previous speed V_{i-1}; the x-axis shows the current speed V_i, discretized with resolution of $1m/s$;

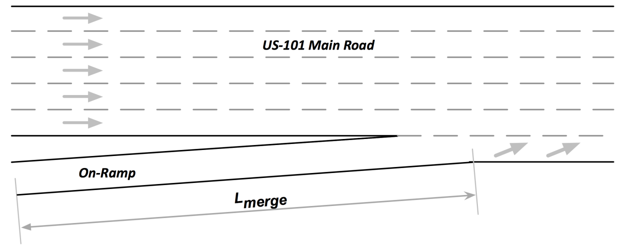


Fig. 4: On-ramp from Ventura Boulevard to US-101 freeway. Since the on-ramp has an auxiliary lane, it is hard to precisely locate the merging point. However, from the US-101 dataset, we know that the average merging distance among all merging cars is about 90 meters. That is equivalent to have a 90-meter merging ramp.

and the y-axis shows the probability of a particular speed transition occurring. If the intention is "Yield", there is higher probability to decrease speed. For example, in Fig. 5c, the black line is higher than the green to the right of the dashed line (previous speed), which means if the merge car driver decides not to yield to the host, it has higher probability to accelerate; on the other hand, the black line is lower than the green to the left of the dashed line, which means the merge car is more likely to decelerate if it decides to yield to the host. These results are consistent with intuitions. Additionally, each speed has specific transition probabilities under different intention. Those transition probabilities are not necessary Gaussian or of other parametric forms, unlike iPCB [4] which only assumes Gaussian distribution to characterize the probabilities of all speed changes.

B. Time Model

The Time model can also be learned from the dataset. The task is to estimate $P(T_m, T_h|I)$. Similar to the speed transition model, the time-to-arrival distribution is also divided into two classes: 1) Yield; 2) Not Yield. Fig. 6a and Fig. 6b show these transition models, where the x-axis is the merging car's time-to-arrival T_m and the y-axis is the host car's time-to-arrival T_h. the z-axis reflects the probability of observing such a pair of times-to-arrival under the Yield or Not Yield intention hypothesis.

V. EXPERIMENTAL RESULT

We conducted two sets of experiments in simulation: 1) reacting to merging vehicles with real-data trajectories which are extracted from datasets; and 2) reacting to merging vehicles which use a manually designed motion strategy. We perform the second set of tests to evaluate the generality of our method with respect to differing merging car strategies and a broader range of initial conditions (speeds and relative location.) It should be emphasized that even though we programmed the strategy of the merging car in the second experiment, the host car does not know the strategy or true intention of the merging car. All the host car can do is observe the state of the merging car and estimate its intention by using our model. We compare our new algorithm with the following methods:

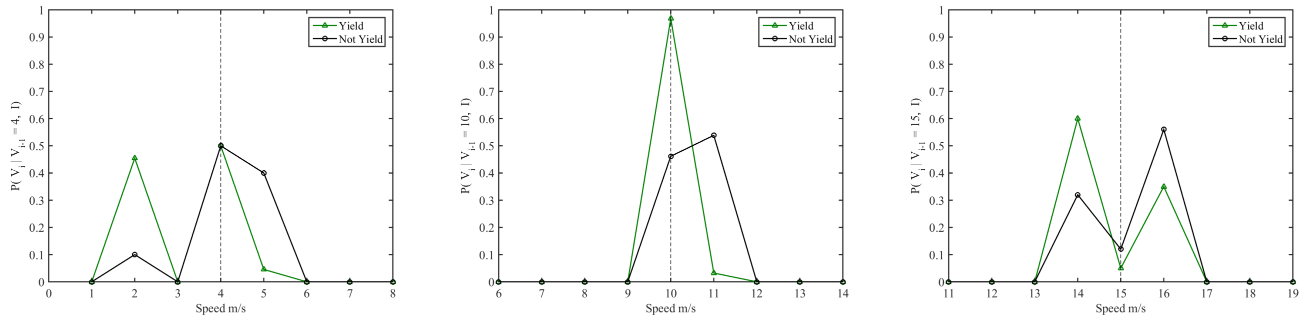
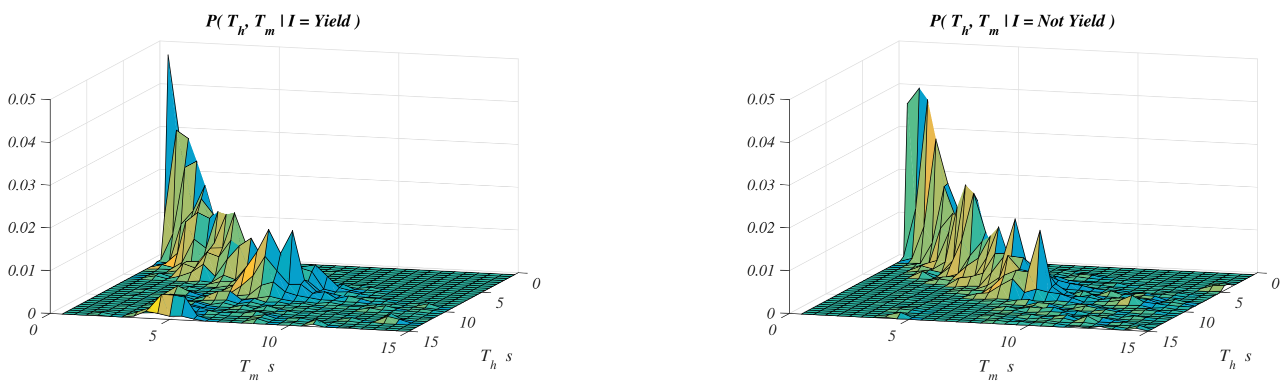


Fig. 5: Examples of speed transition probability $P(V_t|V_{t-1}, I)$ which are learned from training data. The vertical dashed line is the previous speed; the x-axis indicates the current speed; Two colors indicate different intentions. Note that each speed has different transition probabilities under different intentions.



(a) $P(T_m, T_h|I = \text{Yield})$. Merge car yields the host.

(b) $P(T_m, T_h|I = \text{Not Yield})$. Merge car does not yield.

Fig. 6: Time-to-arrival transition probability distributions which are trained from the dataset.

1. *ACC merging*, a non-cooperative method that distance-keeps to the merging car if it is closer to the merging point;
2. *Slot checking*, which is adopted from the Urban Challenge [1].
3. *iPCB*, which is proposed in [4].
4. *PGM-G*, which uses the proposed PGM structure, but assumes Gaussian Distribution on the speed transition probability, like iPCB [4].

We use two criteria to verify the performance of these algorithms: 1) failure rate based on number of collision scenarios; 2) average computation time (for successful cases only). The first criterion deals with safety; the second with efficiency. Vehicles on the main road and ramp have the same task: they should cooperate to merge together safely and efficiently. Therefore, regardless of the main/ramp road geometry, the ramp merging problem is topologically symmetric, so there is no difference if we make the main road or the ramp vehicle autonomous. In our experiments, we put the autonomous vehicle on the main road.

A. Experiments with real data

To validate the learned model on real data, we use the remaining 2/3 of the US-101 dataset and the full I-80 dataset [11] for testing. The setup and traffic conditions are shown in Table I. L_{merge} is the average merging distance on the on-ramp. Since the data were collected during morning rush hour, the average speeds are fairly low. Here we use Space-Mean-Speed (SMS) [12], the total distance traveled by vehicles over the total traveling time of these vehicles, to represent average speed. There is traffic congestion, so those are not typical highway speeds and there is traffic congestion. We do not consider the interaction between adjacent vehicles in the host lane but only cooperation between cars in the host and merging lanes. We only extract merge/host vehicle pairs, and treat them individually to train our model. There are 354 pairs of merging-host pairs in US-101, and 452 pairs in I-80.

TABLE I: Features of the US-101 and I-80 datasets

Dataset	L_{merge} m	SMS [12] m/s (mph)	Num. of Pairs.
US-101	90.4	12.4 (27.7)	354
I-80	110.8	14.2 (31.3)	452

The host car uses PGM for intention estimation and LQR[10] for low-level control, and we apply real trajectories to the merging cars. In a given test, the merge car replays a real trajectory from the dataset, and the host vehicle's

TABLE II: Statistical results for different methods

Method	US-101 Data %	I-80 Data %	Designed Test I %	Designed Test II %	Cycle rate ms
ACC	17.6	16.5	13.3	6.8	0.05
Slot	14.8	10.4	2.4	4.2	N/A
iPCB	19.3	15.8	0.9	1.2	0.20
PGM-G	20.1	11.3	0.9	1.8	0.51
PGM	8.7	7.3	0.4	0.2	0.08

start position and speed are also taken from the dataset. We then run our proposed PGM method to estimate intention of the merging car and apply the LQR control model. The failure rates are shown in Table II in the “US-101 Data” and the “I-80 Data” columns PGM that has the lowest failure rate, and does well on I-80 even though it was trained on US-101. As we indicated in Table I, I-80 and US-101 have similar traffic conditions. I-80 samples have a longer merging ramp, which means that the host vehicle has more time to interact with the merging vehicle, and the controller also has enough time to adjust the vehicle to a certain speed. This may explain why the experimental results on I-80 are slightly better than these on US-101.

Fig. 7 shows an individual example from US-101 testing data, which compares performance among *iPCB*, *PGM-G* and the proposed *PGM* method and real merging/host data. We use station-vs-time (Fig. 7a) and speed-vs-time (Fig. 7b) plots to visualize their performance, where station is the longitudinal distance relative to the merging point. All simulated host vehicles and the real host vehicle start at the same state (speed at around 9.9m/s and location at 170m away from the merging point). The merging car replays the log in the dataset, and starts at about 100 meters away from the merging point and about 10.4 m/s.

In Fig. 7a the horizontal dashed line indicates the merging point. Before reaching that point, the merging car runs on a separate ramp, so the intersections below the dashed line (e.g., the intersection between the green and red line) only indicates that they are the same distance from the merging point, not a collision. However, an intersection above the dashed line, e.g., the green and blue line, does indicates a collision. Therefore, according to Fig. 7a, applying iPCB is problematic, but using PGM avoids collision. According to Fig. 7b, iPCB (blue line) accelerates before 4s and then decelerates until 14s, which means that before the 4th second, iPCB makes improper estimation and after that, it tries to slow down but it has accelerated too much. The PGM curve (orange) in Fig. 7a is the closest one to the ground-truth – the real host trajectory (purple). Even though PGM-G (the red line) can interact without collision, the PGM performs much more similar to human behavior. Since the distance between the PGM-controlled host car and the merging car becomes smaller, and the merging car starts slowing down from the 10th second (according to Fig. 7b), PGM makes the correct decision that the merging car will reach the merging point first and not yield to the host car. Thus the PGM sends deceleration commands which make the orange curve (the PGM) diverge from the purple curve (human driving host car ground-truth)

In sum, the PGM is more conservative than the human driver and makes more space between the two cars, and it performs similarly to human driver at the early stage of merging.

B. Experiments with manually designed merging strategy

To test the generality of our algorithm against a different merging behavior, we applied a manually designed merging strategy. The intention estimation and control part of the host vehicle remained unchanged. We implemented an aggressive strategy for the merging vehicle:

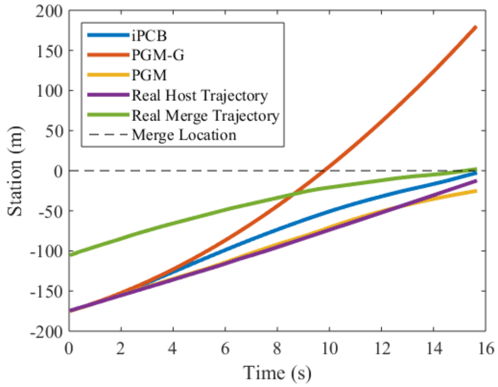
- If no car ahead, accelerate to speed limit.
- If the host car is driving ahead, distance-keep to it.

In the following two tests, after setting up initial states in different ways, the merging car applied our designed strategy, and the host car used the 5 different models including PGM to react.

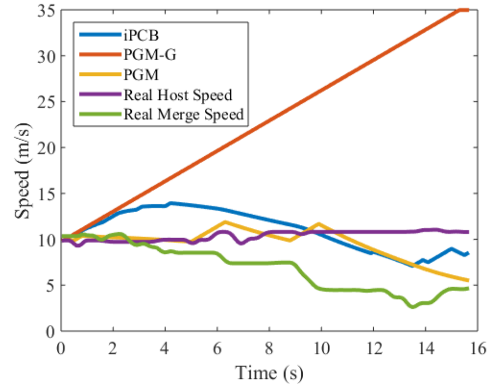
1) Designed Test I: The initial states of the merge and host vehicle were taken from the datasets. This test shows the performance of our method under a different merging strategy. Column “Designed Test I” shows the result. PGM also has the lowest failure rate here. iPCB and PGM-G have the same failure rate, because iPCB relies heavily on accurate merging-car acceleration, which can be easily calculated according to the designed merging strategy. The PGM-G does not perform as well as the PGM, since it also uses the Gaussian assumption which is used in iPCB. iPCB abd PGM-G tend to have similar estimations. As expected, except ACC, the failure rate dramatically dropped compared with the real data tests because when the strategy are implemented in the simulator where there is no noise. The ACC merging is not a cooperative method, so its failure rate does not decrease as much as that of the other, cooperative methods.

2) Designed Test II: Based on the dataset and the road geometry shown in Fig. 4, the origins of the main/ramp roads are set to be 90 meters away from the merging point. The merging car starts at the origin, and the host car's position varies from +5 to -5m at 1m intervals; giving 11 cases of initial distances between two cars. Each car with initial speed varies from $1 \sim 25$m/s at 1m/s intervals, so there are 625 combinations of initial speeds. In total, there are 6875 different cases (Note that this number of cases is almost 9 times of the total cases in US-101 and I-80). This test shows the generality and performance of our method over a broader range of initial states. Column “Designed Test II” shows the result. The PGM still has the lowest failure rate. With the same reason in “Designed Test I”, iPCB and PGM-G have the similar failure rate.

These experiments indicate that our proposed PGM method has some degree of generality, thus it can handle



(a) Station-vs-Time plot of host cars which use different methods for intention estimation. The dashed line indicates merging location, which was set as the zero position.



(b) Speed-vs-Time plot of host cars which use different methods for intention estimation.

Fig. 7: An Example of comparison between *iPCB*, *PGM-G* and *PGM* method.

different types of behaviors and a broader range of initial states (speeds and positions).

VI. CONCLUSIONS

Both real data and designed-strategy tests results show that the proposed method has the lowest failure cases and it improves intention estimation in merging control, compared with previous algorithms. Additionally, its behavior is similar to that of human drivers and somewhat more conservative. Our new approach can enhance the safety of merging ramp behavior of autonomous cars. In the future, we will refine the algorithm by considering how previous intentions affect the current one. We will also extend our method to estimate long-term motion of merging vehicles and take advantage of a broader set of training data.

ACKNOWLEDGMENT

The authors would like to thank Junqing Wei and Wenda Xu who contributed to previous ideas of ramp merging.

REFERENCES

[1] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[2] P. Hidas, “Modelling vehicle interactions in microscopic simulation of merging and weaving,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 1, pp. 37 – 62, 2005.

[3] D. Marinescu, J. urn, M. Bouroche, and V. Cahill, “On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach,” in *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*, Sept 2012, pp. 900–906.

[4] J. Wei, J. M. Dolan, and B. Litkouhi, “Autonomous vehicle social behavior for highway entrance ramp management,” in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 201–207.

[5] J. Wei, J. M. Dolan, J. M. Snider, and B. Litkouhi, “A point-based mdp for robust single-lane autonomous driving behavior under uncertainties,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2586–2592.

[6] J. Pineau, G. Gordon, S. Thrun *et al.*, “Point-based value iteration: An anytime algorithm for pomdps,” in *IJCAI*, vol. 3, 2003.

[7] A. G. Cunningham, E. Galceran, R. M. Eustice, and E. Olson, “Mpdm: Multipolicy decision-making in dynamic, uncertain environments for autonomous driving,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1670–1677.

[8] J. Schlechtriemen, F. Wirthmueller, A. Wedel, G. Breuel, and K. D. Kuhnert, “When will it change the lane? a probabilistic regression approach for rarely occurring events,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 1373–1379.

[9] D. Lenz, T. Kessler, and A. Knoll, “Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search,” in *Intelligent Vehicles Symposium (IV)*. IEEE, 2016, pp. 447–453.

[10] J. M. Snider, “Automatic steering methods for autonomous automobile path tracking,” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-09-08, February 2009.

[11] “NGSIM homepage. FHWA.” 2005-2006. [Online]. Available: http://ngsim.fhwa.dot.gov.

[12] N. J. Garber and L. A. Hoel, *Traffic and highway engineering*. Cengage Learning, 2014.