2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Smooth Behavioral 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 enhance our previous 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 proposed approach inherits our previous Probabilistic grahpical Model (PGM) and distance-keeping framework. 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. The PGM-based intention estimation is followed by an off-the-shelf distance-keeping model to generate proper acceleration/deceleration controls. The PGM plays a plug-in role in our self-driving framework. The new model eliminates two assumptions in the previous model: 1) a fixed merging point for all merging agents, which is hard to determine before the merging vehicles make the merge; 2) Perfect velocity measurement, which requires sophisticated perception systems. We validate the performance of our method both on real merging data and using a designed merging strategy in simulation, and show significant improvements compared with previous methods. Parameter design is also discussed by experiments. The new method is computationally efficient, and exhibits better robustness against sensing uncertainty.**

I. Introduction

Autonomous driving is drawing tremendous attention among automakers and end-users, and some of the technologies, such as Adaptive Cruise Control (ACC), Lane Keeping/ Lane Departure Warning, Collision Detection, and Emergency Brake, have been integrated into normal passenger vehicles. The most advanced products can assume control in specific simple scenarios. For example, GM's Super Cruise production vehicle offers Full Speed Range ACC (Stop & Go ACC) and hands-free driving at freeway speeds on a single lane freeway. Audi's "STOP and GO" adaptive cruise control (ACC) enables 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.

While such features can allow cars to drive hands-free under certain conditions, they do not guarantee proper interaction with other cars in all conditions. Further, for the foreseeable future, autonomous cars need to interact with significant number of human-driven cars. In the presence of Vehicle-to-Vehicle (V2V) communication, some V2V systems may recommend behaviors to human drivers to cooperate with others; however, whether the driver can properly follow the indication given by the system remains uncertain. It is therefore important for autonomous cars to comprehend and exhibit social behaviors to suitably interact with human-driven cars or other autonomous cars. 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. A typical social interaction in driving is ramp merging. Normally, a human driver will implicitly "negotiate" with one or more drivers on a ramp or main road, estimate their intentions, and then makes 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, based on some relevant 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. Unlike our previous method, the current approach also considers an auxiliary lane which follows the on-ramp, instead of using a fixed merging point for all merging cars. Further, we do not assume an accurate measurement of speeds. Instead, a series of locations of the merging targets are tracked and further smoothed in order to obtain an optimal estimate of the dynamics of the merging car. The goal of our method is to estimate whether or not the merging car intends to yield to the host car, and then safely react to it.

II. Related Work

There are several references that address the merging problem. Urmson et al. [1] and Marinescu et al. [2] 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. Lu and Hedrick [3] formulated the merging planning as a platoon control problem. But the algorithm still requires a "coordination layer" to select a pair of main-road cars to interact, i.e., need a decision

*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. chiyud@andrew.cmu.edu

[2]John M. Dolan is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA. jdolan@andrew.cmu.edu

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

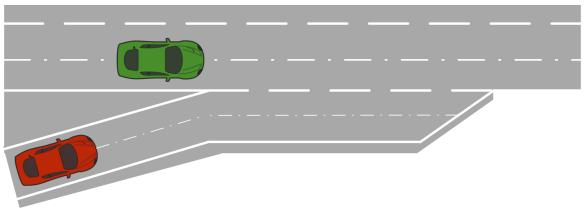

Fig. 1: Merge scenario with an auxiliary lane. The host car (green) is an autonomous vehicle, running on the main road; the merge car (red) is a human-driven car, running on the ramp. The merging ramp has an auxiliary lane which merges into the main road at the end of the ramp. The merging car has to perform the merge and lane changing before reaching the end of the ramp. The host car should behave socially to ensure safety and efficiency.

for yielding or not yielding a host vehicle. 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 single-lane behavior framework which takes uncertainties into account. They also applied a cost function to evaluate and select the proper strategy. Schlechtriemen et al. [6] proposed a learning-based approach to estimate the lane-changing intention. They calculate lane-changing probabilities by vehicles' current lateral speeds using Random-Decision-Forest and Gaussian Mixture Regression. Dong et al. [7] predicted the start/end points of a lane-change behavior of a target vehicle given its surrounding vehicles' trajectories, by using a non-parametric learning model. Lenz et al. [8] 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 on massive real world data. Liu et al. [9] generate a cooperative longitudinal speed profile along with a pre-planned path. A set of quadratic programs are used to approximate the non-convex temporal optimization, which are solved by the slack convex feasible set (SCFS) algorithms. Nilsson et al. [10] formulated cooperative planning as an optimization problem under a Model Predictive Control (MPC) framework. The weighted effects of acceleration and braking are optimized subject to the trajectory's shape and feasibility. The author provides a straightforward way to transform the problem into a well-defined optimization problem that can be solved by applying a specific solver. However, the manual tuning of weights is difficult. Also, the equation to be optimized and objective functions are designed and hand-tuned, without the use of data. These works provide a feasible way to formulate and perform planning in dynamic environments, but both methods rely on an oracle path planner to pre-generate a cooperative path. Neither approach explicitly solves the prediction problem, i.e., estimating other vehicles' behaviors.

Zhan et al. [11] propose an integrated decision-making and planning framework under uncertainty, to achieve a driving strategy which is defensive to real threats, but not conservative to threats of low probability interaction is not considered. Galceran et al. [12] utilized past data and extended the reaction ability of autonomous cars from a single lane to multiple lanes, including merging. They modeled the multipolicy decision-making into a partially observable Markov decision process (POMDP). To speed up the evaluation, a limited number of actions (policies) are used, such as "change-lane-left/change-lane-right" and "lane-normal". To avoid discrete states, Bai et al. [13] proposed a continuous-state POMDP using a belief tree and the model was applied to navigating intersections. However its actions are discrete and represented by a generalized policy graph (GPG). Seiler et al. [14] proposed an online and approximate solver for a continuous action POMDP, but only tested in toy problems.

In this paper, we instead use a probabilistic graphical model (PGM) to describe dependency among observed data and estimate other cars' intentions. The task of the PGM is to generate an intention estimate with maximum probability, given observed information. The PGM clearly organizes relationships among short-term historical data and intention estimation. Using the properties of the dependencies of nodes in the graphical model, 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. In addition, the model does not assume a perfect perception system; instead, it takes direct measurement of positions, and uses filtering and smoothing methods to obtain an optimal estimate of latent state, i.e., locations and speeds. Although in this paper, we focus on ramp merging, the intention estimation can also be extended to various cooperation situations, such as lane changing and stop sign traversal.

III. PGM-BASED INTENTION ESTIMATION

A. Structure of PGM

In our previous methods [15], [16], speed information plays an important role for intention estimation. However, there are few sensors that can directly output accurate speed measurements. For instance, Radars can measure an object's range rate, which is only considered valid if the object is running right in front of or behind the sensor. Otherwise, the range rate is the projected speed along the direction between the sensors and the object. To avoid the limitations of existing speed sensor and make the system more feasible for commercial application, instead of directly sensing speeds, we keep an array of location observations, and smooth out speeds as well as locations (having actual speeds can further enhance estimation). In the graphical model (Fig. 2), only

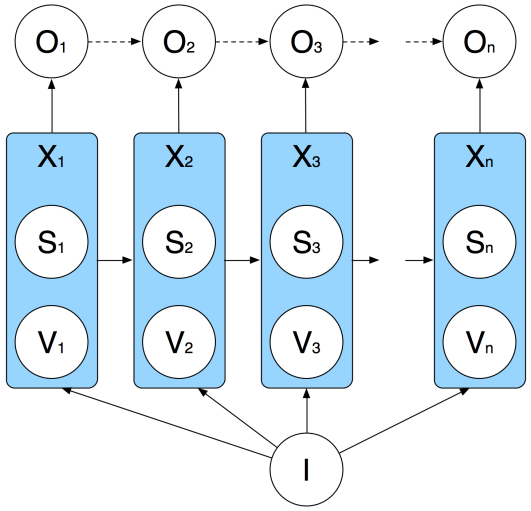

Fig. 2: The graphical model of ramp merging considering position observations and intentions. O nodes are positions for the merging car relative to the end of the auxiliary lane, and are the direct output of sensors; X is the latent state estimated by smoothing and it consists of S, V nodes for positions and speeds; The I node is the intention. The number of required observations and states is determined based on experiment.

location nodes O_i are observable. The observation is the relative location of the target with respect to the end of the merging ramp; I is the intention node; latent nodes X_i are the state of the merging car, which is expected to have an optimal estimation. It consists of nodes S, V: S_i are the true locations; V_i nodes are speeds after smoothing. Compared with the model in [15], [16], the proposed model does not include Time-to-Arrival nodes, since their information can be derived from location and speed nodes.

B. Estimating intentions from observations

Based on the observations, the model is expected to output probability of intentions, i.e.,:

$$P(I|\mathbf{O}, \mathbf{X}) \tag{1}$$

where I is the intention and $\mathbf{O}, \mathbf{X}$ are arrays of observations $\{O_i\}$, state $\{X_i\}$ which contains the estimated distances to the merging point $\{S_i\}$, and estimated speeds $\{V_i\}$. According to the structure of the graphical model, I and $\mathbf{O}$ are conditionally independent given $\mathbf{X}$. Therefore, (1) yields the same probability which is stated in [15]:

$$P(I|\mathbf{X}) \tag{2}$$

From the graphical model, the state X_i has the Markov Property given intention. Thus (2) can be further simplified by assuming $\mathbf{X}$ to be multi-variable and uniformly distributed:

$$\begin{aligned} P(I|\mathbf{X}) &\propto P(\mathbf{X}|I)P(I) \\ P(\mathbf{X}|I) &= P(X_1, X_2, ..., X_n|I) \\ &= P(X_1|I)P(X_2|X_1, I)...P(X_n|X_{n-1}, I) \end{aligned} \tag{3}$$

Since the initial state X_1 can vary regardless of the intention, here we assume X_1, I are independent, thus $P(X_1|I) = P(X_1)$. Since there is no preference for X_1, it can be assumed to have a uniform distribution, namely $P(X_1) = \alpha$. To prevent underflow, log-likelihood is used:

$$\log P(\mathbf{X}|I) = \alpha \sum_{i=2}^{n} \log P(X_i|X_{i-1}, I) \tag{4}$$

The estimated intention is:

$$I^* = \arg\max_I \ \log P(I|\mathbf{X}) \tag{5}$$

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

C. Smoothing from the observations

Instead of measuring the speeds, the model relies on location measurements, and then derives speeds using filtering and smoothing. As typically the locations cannot be perfectly measured from sensors, it is necessary to estimate their true values. Since the model collects a certain number of location observations, all of these data can be utilized to smooth the estimation of true values, i.e., estimating $X_i = (S_i, V_i), 1 \leq i \leq n$ using a series of observations O_i.

$$P(X_i|O_{1:n}) \tag{6}$$

According to [17], [18] the predicted distribution of time step *i+1* by adapting Bayesian optimal smoothing is:

$$P(X_{i+1}|O_{1:i}) = \sum_{X_i} P(X_{i+1}|X_i)P(X_i|O_{1:i}) \tag{7}$$

And the smoothing estimation of the state at any time step $i < n$ is:

$$P(X_i|O_{1:n}) = P(X_i|O_{1:i}) \sum_{X_i} \frac{P(X_{i+1}|X_i)P(X_{i+1}|O_{1:n})}{P(X_{i+1}|O_{1:i})} \tag{8}$$

With a proper edge condition, i.e., $P(X_n|O_{1:n})$ which will be discussed in the following paragraphs, every past state can be solved by iteratively evaluating (8).

Here, we assume the distribution of the state at any time step given that the full observation is a Normal Distribution, i.e.,

$$X_i|O_{1:n} \sim N(X_i|\mathbf{M}, \mathbf{P}) \tag{9}$$

Then the Bayesian smoothing equations in (7) and (8) become a *Rauch-Tung-Striebel smoother* (RTS smoother, which is also called the Kalman smoother). The speed part of the maximum *a posteriori* of the state will be taken as the condition of the intention I's estimate. The RTS smoother consists of two steps:

1) The Forward Prediction: The Forward Prediction step using a Kalman Filter to update the mean M_i and the covariance P_i of the distribution, as well as the Kalman Gain G_i:

$$\begin{aligned} M_{i+1}^- &= AM_i, \\ P_{i+1}^- &= AP_{i+1}A^T + Q, \\ G_i &= P_i A^T [P_{i+1}^-]^{-1} \end{aligned} \tag{10}$$

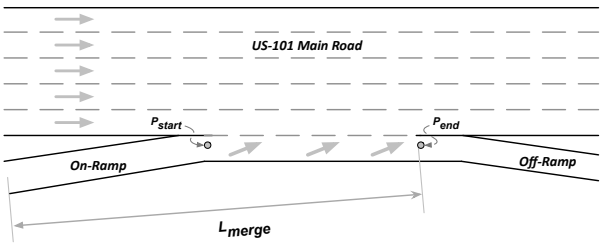

Fig. 3: On-ramp from Ventura Boulevard to US-101 freeway with an auxiliary lane (the lane with gray arrows in the figure). Merging vehicles can merge onto the main road between P_{start} and P_{end} in the auxiliary lane. P_{end} is used for the reference point with which the relative distances can be calculated.

where A is the motion model of the Kalman Filter, which will be specified in the following section. Q is the covariance for the process noise,

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} \frac{q\Delta t^3}{3} & \frac{q\Delta t^2}{2} \\ \frac{q\Delta t^2}{2} & q\Delta t \end{bmatrix}$$

2) The Backward Recursion:

$$\begin{aligned} \hat{M}_i &= M_k + G_k[\hat{M}_{i+1} - M^-_{i+1}], \\ \hat{P}_i &= P_i + G_k[\hat{P}_{i+1} - P^-_{i+1}]G^T_k \end{aligned} \tag{11}$$

where the $\hat{\cdot}$ indicates the smoothed variables. The backward equations start from the last step, where $\hat{M}_n = M_n, \hat{P}_n = P_n$. Note that the result $\hat{\mathbf{M}}$ contains all optimal states, which is an array of location and speed tuples, $X = (S, V)$.

D. Intention Estimation Procedure

The Smoothing procedure (Equation. 8, 11) will be called before every evaluation step (Equation. 5), with a new observation received and the earliest observation dropped. If $I^* =$ **Not Yield**, which indicates that the merging vehicle tends to speed up and reach the merging point before the host vehicle. Then the host vehicle activated a distance-keeping model to keep a desired safe longitudinal distance to the merging car that runs on the ramp. On the other hand, if $I^* =$ **Yield**, the host car will ignore the merging car and accelerate to the speed limit. Details refer to Alg. 1 in [15].

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 [19] to extract a model of cooperative behavior between host and merging vehicles. The dataset was obtained from overhead cameras near the US-101 Ventura Boulevard entrance ramp in the Los Angeles area, as shown in Fig. 3. Another similar dataset was obtained by the same procedure on Eastbound of I-80 around San Francisco Bay Area in Emeryville, CA.

Cars in this region were recorded 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 bottom 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. The goal of training is to estimate the conditional probability of intentions given historical state information, i.e., $P(I|X_{1:n})$. The two classes of data are used to train two different models, i.e., speed transition probability distributions. We do not assume those transition probabilities to be necessarily Gaussian or of other parametric forms, unlike iPCB [4] which assumes a Gaussian distribution to characterize the probabilities of all speed changes. Note that there is no conflict between the transition probabilities here and the Gaussian distribution assumption in the RTS smoother (7). The transition models serve as a probabilistic "control" model, whereas the noise in the RTS prediction model is the uncertainty of the next state given a control. SmoothPGM does not fully trust the speed observation, which is different from the vanilla version of PGM in [15]. Instead it also considers the location changes and obtains the speed information by an optimal estimation.

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:

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 a Gaussian Distribution for the speed transition probability, like iPCB [4].
5. *PGM*, which is proposed in [15].
6. *smoothPGM*, which is proposed in this paper.

We use three criteria (as shown in Table II) to verify the performance of these algorithms: 1) failure rate based on number of collision scenarios; 2) average minimum distance with the host and the merging car when the host passes the

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

Dataset	Ramp Length m	SMS [20] m/s (mph)	Num. of Pairs.
US-101	640	12.4 (27.7)	354
I-80	503	14.2 (31.3)	452

TABLE II: Statistical results for different methods

Method	US-101 Data		I-80 Data		Designed Test I	Designed Test II	Cycle rate
	%	D(m)	%	D(m)	%	%	ms
ACC	17.6	22.2	16.5	7.0	13.3	6.8	0.05
Slot	14.8	22.8	10.4	10.3	2.4	4.2	N/A
iPCB	19.3	23.7	15.8	13.7	0.9	1.2	0.20
PGM-G	20.1	24.3	11.3	14.6	0.9	1.8	0.51
PGM	8.7	25.8	7.6	16.4	0.4	0.2	0.08
smoothPGM	3.6	25.5	5.3	17.4	0.3	0.2	0.08

merging point; 3) average computation time (for successful cases only). The first two criteria deal with safety; the third with efficiency. Vehicles on the main road and ramp have the same task: they should cooperate to merge together safely and efficiently. In our work, we initially put the autonomous vehicle on the main road, and expect proper cooperative driving w.r.t. the merging-in vehicles from the on-ramp.

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 [19] for testing. Obviously, no collisions occur in the real data. The main idea for this test is to compare the performance of the new method with that of previous methods. None of these autonomous methods are as capable as a human driver, but we expect to improve upon them. The setup and traffic conditions are shown in Table I. Ramp Length here is the length of the on-ramp plus the auxiliary lane (distance from P_{start} to P_{end} in Fig. 3). Since the data were collected during morning rush hour, the average speeds are fairly low. Here we use Space-Mean-Speed (SMS) [20], 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.

The host car uses each of the methods in the "Method Column" for intention estimation and LQR for lower-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 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. smoothPGM has the lowest failure rate, and does well on I-80 even though it was trained on US-101. In tests of different datasets, the D(m) columns show average distances between the host vehicle and the closest merging car when the host passes the merging point. smoothPGM has the highest average distance (it has similar D(m) with PGM), which is an indication of its greater safety. There are two reasons that make smoothPGM outperform our previous approach (PGM), which are highly related to the design of the proposed model: 1. smoothPGM does not rely on a fixed merging point. Since the merging point of PGM is fixed and it ignores the effect of the auxiliary lane, there are merging cars which run beyond the assumed merging point, resulting in a negative or undesired measurement (distance-to-merge); 2. The proposed method smooths out the speeds from location measurements instead of directly trusting noisy speed measurements.

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. smoothPGM 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 and smoothPGM, since it also uses the Gaussian assumption which is used in iPCB. iPCB and PGM-G tend to have similar estimations. As expected, except for ACC, where the failure rate dramatically dropped compared with the real data tests because the strategies are implemented in the simulator and 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. 3, 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 cases is almost 9 times of the total cases in US-101 and I-80). This test

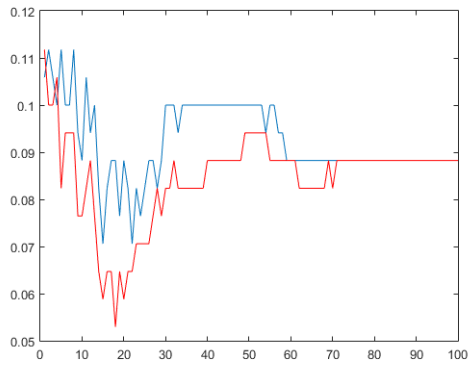

Fig. 4: Collision rates vs. different numbers of speed nodes with (red line) and without smoothing (blue line).

shows the generality and performance of our method over a broader range of initial states. Column "Designed Test II" shows the result. The smoothPGM and PGM still have the lowest failure rates. For the same reasons as for "Designed Test I", previous approaches have similar failure rates as the proposed method. These experiments indicate that our proposed smoothPGM method has some degree of generality, thus it can handle different types of behaviors and a broader range of initial states (speeds and positions). However, there are still some collision cases that cannot be handled by the proposed algorithm. The algorithm is thus not as safe as a human driver, and requires further improvement to obtain zero collision rate.

C. Collision Rates vs. Number of Speed Nodes

To determine the proper number of state nodes, and inspect the smoother's effect on the estimation result, the proposed method was applied to the dataset with varying number of speed nodes and compared with a version which does not have the smoothing step. In Fig. 4, the blue line is the result for non-smoothing PGM, and the red one is for the proposed method. Since more past information can help the estimation, both lines go down as we increase the number of state nodes to around 20, and then go up since redundant nodes affect the sensitivity to the present dynamic changes. For the same reason, the two lines converge to one collision rate. Furthermore, the red line is below the blue line and its minimum is lower than that of the blue one. This plot indicates that by using the new approach, one can achieve the same level of collision rate by using fewer state nodes than the previous method, and also obtain a lower collision rate with no more than 20 state nodes.

VI. CONCLUSIONS

Both real data and designed-strategy test results show that the proposed method has the lowest failure rate and it improves intention estimation in merging control, compared with previous algorithms. Additionally, it lowers the demands on the perception system by solely using location information with smoothing. In the future, we will learn the dependency among intentions and physical measurement from data and experiments. We will also extend our method to estimate long-term motion of merging vehicles, in the scenarios that have more than one merging car, and take advantage of a broader set of training data.

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] 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.

[3] X.-Y. Lu and K. Hedrick, "Longitudinal control algorithm for automated vehicle merging," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 1. IEEE, 2000, pp. 450–455.

[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. 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.

[7] C. Dong, Y. Zhang, and J. M. Dolan, "Lane-change social behavior generator for autonomous driving car by non-parametric regression in reproducing kernel hilbert space," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4489–4494.

[8] 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.

[9] C. Liu, W. Zhan, and M. Tomizuka, "Speed profile planning in dynamic environments via temporal optimization," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2017, pp. 154–159.

[10] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh, "Longitudinal and Lateral Control for Automated Yielding Maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1404–1414, may 2016.

[11] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 459–464.

[12] E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction," in *Proceedings of Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015.

[13] H. Bai, D. Hsu, and W. S. Lee, "Integrated perception and planning in the continuous space: A POMDP approach," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1288–1302, 2014.

[14] K. M. Seiler, H. Kurniawati, and S. P. N. Singh, "An online and approximate solver for POMDPs with continuous action space," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 2290–2297.

[15] C. Dong, J. M. Dolan, and B. Litkouhi, "Intention estimation for ramp merging control in autonomous driving," in *2017 IEEE 28th Intelligent Vehicles Symposium (IV'17)*, Jun. 2017, pp. 1584 – 1589.

[16] ——, "Interactive ramp merging planning in autonomous driving: Multi-Merging leading PGM (MML-PGM)," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2017, pp. 2186–2191.

[17] G. Kitagawa, "Non-gaussian smoothness prior approach to irregular time series analysis," in *Adaptive Systems in Control and Signal Processing 1986*. Elsevier, 1987, pp. 303–308.

[18] S. Särkkä, *Bayesian filtering and smoothing*. Cambridge University Press, 2013, vol. 3.

[19] "NGSIM homepage. FHWA." 2005-2006. [Online]. Available: http://ngsim.fhwa.dot.gov.

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