2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

A Point-based MDP for Robust Single-Lane Autonomous Driving Behavior under Uncertainties

Junqing Wei*, John M. Dolan†, Jarrod M. Snider‡, Bakhtiar Litkouhi§
*Department of Electrical and Computer Engineering,
Carnegie Mellon University, Pittsburgh, PA 15213 USA
†The Robotics Institute and Department of Electrical and Computer Engineering,
Carnegie Mellon University, Pittsburgh, PA 15213 USA
§GM-CMU Autonomous Driving Collaborative Research Lab,
GM R&D Center, Warren, MI 48090 USA

***Abstract*—In this paper, a point-based Markov Decision Process (QMDP) algorithm is used for robust single-lane autonomous driving behavior control under uncertainties. Autonomous vehicle decision making is modeled as a Markov Decision Process (MDP), then extended to a QMDP framework. Based on MDP/QMDP, three kinds of uncertainties are taken into account: sensor noise, perception constraints and surrounding vehicles' behavior. In simulation, the QMDP-based reasoning framework makes the autonomous vehicle perform with differing levels of conservativeness corresponding to different perception confidence levels. Road tests also indicate that the proposed algorithm helps the vehicle in avoiding potentially unsafe situations under these uncertainties. In general, the results indicate that the proposed QMDP-based algorithm makes autonomous driving more robust to limited sensing ability and occasional sensor failures.**

I. Introduction

Starting in the 1980s, autonomous driving has gradually become a fast-developing and promising area. Autonomous driving technology has the ability to provide driver convenience and enhance safety by avoiding some accidents due to driver error. Though autonomous vehicles have already shown the ability to perform distance keeping, lane-keeping, intelligent route planning, off-road navigation, and interact with human-driven urban traffic, their robustness and intelligence are still not sufficient for real traffic environments. Among the range of autonomous driving scenarios, single-lane freeway autonomy has relatively low complexity and does not impose a high driver workload. Therefore, building autonomous vehicles with reliable and powerful single-lane freeway autonomy that can perform robustly under practical constraints (e.g. sensor limitations, potential failures, system delays) is an important milestone in reaching the goal of fully autonomous driving.

Since the late 90s, car manufacturers have gradually equipped their vehicles with Adaptive Cruise Control (ACC) systems that can perform distance keeping at high speeds [1]. In the past few years, full-range ACC, which operates over the speed range from stop-and-go to high-speed driving, has also appeared on the market [1]. However, ACC has limited ability in reducing the driver workload, since drivers still need to constantly supervise the steering control and monitor the vehicles distance keeping behavior.

In 2007, the DARPA Urban Challenge provided researchers in the autonomous driving field the means to test the latest sensors, computer technologies and artificial intelligence algorithms, instrumented and implemented on autonomous vehicles, in a set of emulated urban driving scenarios [2]. Basic interaction between autonomous vehicles and human-driven vehicles was proven in low-density, low-velocity traffic. In the competition, most autonomous vehicles' behaviors are controlled by a set of manually designed rules [3], which focused more on implementing the functionality rather than achieving high driving performance. The autonomous pilot's driving therefore did not achieve the same level of comfort and robustness as that of human drivers.

To improve the performance of the autonomous pilot, a Prediction- and Cost function-Based (PCB) algorithm for autonomous vehicle reasoning was proposed and tested [4], [5]. Instead of an if-else based reasoning system, this cost function-based approach improved the robustness of the autonomous distance keeping, lane selecting and merge planning. Four categories of cost function were built for autonomous driver decision making. A learning mechanism for PCB was also developed to automatically adjust its cost function weights and configuration by learning from human drivers behaviors [6]. There are also other algorithms used for single-lane autonomous driving, such as Model Predictive Control (MPC)-based cruise control [7] and neural network-based autonomous pilot developed in [8], [9]. However, most state-of-the-art autonomous behavior control algorithms [5], [6], [8], [9] are based on a deterministic reasoning model, which overlooks the potential effects of certain uncertainties.

In autonomous driving, there are considerable uncertainties in both the environment (e.g. traffic situations) and the vehicle itself (e.g. perception noise) [10]. The inability to properly adjust the vehicle's behavior to different uncertainty levels may lead to improper driver decisions. For instance, at some intersections, human drivers will slow down if they cannot get a clear view of vehicles coming from a particular direction, whereas some autonomous vehicles may assume the intersection is empty based on their perception input. This inability to model the limitations of the robot's perception system will affect the reliability of an autonomous driving system.

A Markov Decision Process (MDP) is a cost-driven predic-

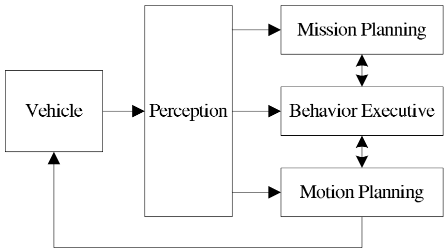

Fig. 1. Autonomous Driving Platform Framework

tive control algorithm and a useful tool for decision making [10]. MDP is able to take the uncertainty of state transitions into account. The Partially Observable Markov Decision Process (POMDP), an extension of MDP, is a powerful but computationally expensive tool to handle perception system uncertainties using belief space, which is a probabilistic representation of the current state [10], [11]. A point-based Markov Decision Process (QMDP), which uses a sampling mechanism to reduce computational expense, can more efficiently find an approximate solution of a POMDP [11]. It takes the uncertainty into account at the first planning step, and then assumes the observations in following steps are deterministic. QMDP has proven to be promising for mobile robot indoor navigation under position uncertainties [12], and is applied here to autonomous freeway driving.

II. Modeling Driving Behavior Control as a Markov Decision Process

A. Behavior layer of Single Lane Autonomy System

There are four primary subsystems in our single lane autonomous driving system, as shown in Figure 1. The perception system analyzes real-time data input from LIDAR, radar and GPS and other sensors. Mission planning optimizes the path to achieve different checkpoints considering the arrival time, distance, or different required maneuvers. The behavior executive system makes tactical driving decisions on such things as distance keeping, lane changing, and neighboring vehicle interactions. Motion planning generates the desired trajectory of the vehicle considering the dynamic parameters and outputs steering and throttle commands. In this paper, we focus on the behavior layer of the system. The objective is for the vehicle to make robust decisions under uncertainties associated with sensing limitations, delays, and actions of surrounding traffic.

B. Basic Markov Decision Process Model

The Markov Decision Process is a cost-driven decision making framework. In the single autonomous driving problem, the state vector X represents the microscopic traffic scenario, defined in Equation 1,

$$\bar{X} = [d_{lead}, v_{lead}, v_{host}, a_{host}, j_{host}]^T \tag{1}$$

where d_{lead} is the distance to the leading vehicle, v_{lead} is the velocity of the leading vehicle, and v_{host}, a_{host} and j_{host} are the speed, acceleration and jerk of the host vehicle. The action set $\bar{A}A$ consists of all candidate output commands, which is a series of different possible acceleration commands in this controller. The motion model is used to predict a series of microscopic traffic scenarios in the next few seconds given a certain output command, as shown in Equation 2,

$$X\bar{X}^i = predict(X_0, strategy^i) \tag{2}$$

where $X\bar{X}^i$ is the vector of prediction outputs at all time steps, as in Equation 3.

$$X\bar{X}^i = [X_0^i, X_t^i, X_{2t}^i, \cdots] \tag{3}$$

The motion model predicts the microscopic traffic scenarios based on the assumption that all cars tend to maintain their velocity but will brake if they are too close to their leaders. The cost corresponding to each strategy $C_{strategy}$ is computed using Equation 4,

$$C_{strategy_i} = \sum_{t=0}^{t_{PredictHorizon}} C_{X(i,t)} \tag{4}$$

in which $t_{predictHorizon}$ is the length of the prediction of the motion model, and $C_{X(i,t)}$ is the cost of each predicted scenario X. Then, the MDP will output the strategy in $\bar{A}A$ with the lowest cost to be executed by the robot.

C. Building Cost Functions

A human-understandable, heuristic and configurable cost function library is used to compute the cost and evaluate scenario $C_{scenario}$ or C_X. Four different costs are built to evaluate the traffic scenario in distance keeping.

- Progress cost: The progress cost represents how well a strategy does in finishing a given task by penalizing those strategies which take longer to finish the task. The goal of the distance keeper is to keep a desired distance $d_{desired}$ to its leader, which is computed in Equation 5, in which v is the current velocity of the vehicle, μ_0 is the distance to the leader when the vehicle is static, and μ_1 is the gain of the desired distance increase corresponding to v.

$$d_{desired} = \mu_0 + \mu_1 v \tag{5}$$

- Comfort cost: While driving a car, in general, human drivers will try to avoid large accelerations for greater comfort. Therefore, a comfort cost Ccomfort is built to represent this logic.
- Safety cost: The safety cost of a scenario consists of two terms: the clear distance cost and the braking distance cost. The clear distance cost $C_{distance}$ penalizes moving too close to surrounding vehicles. However, this cost is not informative enough for us to avoid collision in some situations, since it does not consider the vehicles' velocities. Therefore, another safety cost based on the braking distance difference Δd_{brake} between two vehicles is also considered.
- Fuel consumption cost: The fuel consumption cost is proportional to the fuel usage as estimated by the CMEM statistical model [13].

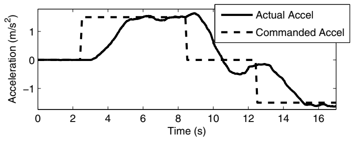

Fig. 2. Step response of the autonomous vehicle

The parameters and the shapes of these cost functions are selected based on case tests and statistical tests in a simulator with simulated traffic vehicles [5]. As shown in Equation 6, the total cost of a scenario is the weighted sum of all these costs.

$$C_{scenario} = \mu_2 C_{progress} + \mu_3 C_{comfort} + \mu_4 C_{safety} + \mu_5 C_{fuel} \tag{6}$$

Then the best distance keeping strategy is selected based on the lowest accumulated cost, which is computed using Equation 4. The performance using this cost function set was verified both in case tests of six manually built scenarios and a statistical test on a long simulated road with randomly generated simulated traffic [5].

Though cost functions based on human domain knowledge led to a functional autonomous driver, it is desirable to achieve single lane driving similar to that of human drivers. This should make autonomous vehicles more predictable, and easier to interact with, by human drivers. Therefore, learning using the mechanism in [6] was implemented to finalize the cost functions parameters by learning from human driver behaviors.

D. Autonomous Driving System Latency Compensation

A fully autonomous vehicle needs some amount of time for sensing and decision making. To achieve better performance, the total system latency needs to be considered. As shown in Figure 1, the vehicle platform uses a multi-layered controller framework. Therefore, the latency from sensing a scenario to the vehicles reaction is the sum of the delays in each layer. Adding the execution period of the sensor, perception algorithm task, behavior control, motion planning and drive-by-wire system together, the latencies or the pure-time-delay (PTD) of the system, is around 0.34s.

For the behavior layer, there is no need to consider the complicated vehicle dynamic model. However, a basic model of the transmission from the desired acceleration command to the actual vehicle acceleration is desirable for better behavior control. Therefore, the execution of the behavior layers distance keeping command is modeled as a first-order system.

To verify the theoretical estimation of the system latency and find out the parameters of the delay model, a road test was performed. In the test, the behavior layer outputs a step desired acceleration command. The step and response are shown in Figure 2.

From the test, it is found that $PTD = 0.45$ and $t_{const} = 1.2$ for the first-order system is.

The MDP has the feature of model predictive controllers that if the system latency is properly modeled in the state transition, it can be well compensated for without modifying other parts of the algorithm [14]. Therefore, the delay model is added to the motion model of the MDP. After modeling the system latency, instead of planning from the current state X, the MDP actually makes decisions based on an estimated state $X(t + t_{PTD}$. The system latency is then compensated for by taking the best action considering the amount of pure time delay. With proper model and compensation of delay, the total cost of the MDP planner defined in Equation 6 decreased by 22% in the test of following the leading vehicle with the standard velocity profile, which including stop-and-go, normal driving and emergency stop situations as shown in Figure 3.

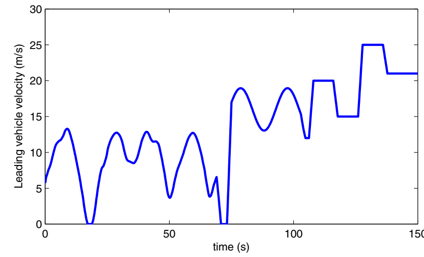

Fig. 3. Standard test velocity profile including scenarios of stop-and-go, high/low speed normal driving and emergency stop

TABLE I
TOTAL COSTS OF USING DIFFERENT STRATEGY SET IN MDP

	$t_{pdt} = 2.4s$	$t_{pdt} = 5.0s$	$t_{pdt} = 12.0s$
$t_{ctr} = 2.4s$	3.60×10^6	2.32×10^6	1.56×10^6
$t_{ctr} = 5.0s$	N/A	3.16×10^6	1.64×10^6
$t_{ctr} = 12.0s$	N/A	N/A	2.44×10^6

III. EXTENSIBLE AND EFFICIENT MDP SOLVER

A. Candidate Set

To ensure the algorithms extensibility and compatibility with non-linear and non-convex cost functions, a search-based MDP solver is used. Because of the constrained computational power and time, only a limited number of candidate MDP strategies can be searched and evaluated in each control cycle. Therefore, the candidate set selection affects the driving performance.

In previous research [5], the candidate strategies were$t_{control} = 4$ seconds long and of constant acceleration. However, this candidate strategy set does not have a prediction horizon long enough to achieve sufficiently high stability and performance. Similar to Model Predictive Control [14], an additional prediction horizon with zero control output is inserted after the original control horizon. The candidate strategies become 'keep constant acceleration for t_1 seconds, then keep constant velocity for another t_2 seconds'. The results of using different candidate sets in a scenario, in which the autonomous vehicle is catching up to its leader, are shown in Figure 4.

A more statistical and comprehensive velocity profile, as shown in Figure 3, is also used to evaluate the performance of different strategy candidate sets, shown in Table I. Based on the experiments using different control and prediction

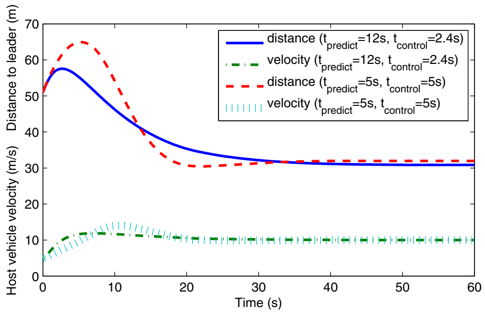

Fig. 4. Performances of using different strategy set in MDP

TABLE II
TOTAL COSTS OF USING DIFFERENT VARIANT-GRANULARITY PREDICTION AND EVALUATION PARAMETERS

	$t_{step}(ms)$	N_{eval}	$Cost$
C-Granularity ($t_{step} = 0.2s$)	1338.2	1740	1.56×10^6
C-Granularity ($t_{step} = 3.0s$)	271.2	135	1.57×10^6
V-Granularity	269.2	135	1.48×10^6

horizons, suitable time horizon values for the candidate set are $t_1 = 2.4s$ and $t_2 = 12s$.

B. Variable-granularity Prediction and Evaluation

For smoother control, the time step of the MDP controller is set to 0.2 sec. The scenario cost is computed at each time step. Therefore, for a prediction horizon of 10 seconds, the evaluation will be called 50 times, which significantly slows down the planner. Though the selection of strategies is based on the total costs of the 50 prediction steps, the further-out predictions have lower confidence. Therefore it is not necessary to use very small time steps to do predictions a few seconds away, and a variable-granularity prediction and evaluation with finer steps at the beginning and larger steps at the end can be used to reduce the computational expense. Table II shows the overall cost using different parameters for descritization. In the experiment, the vehicle is making real-time decisions based on the variable-granularity methods. The total costs in Table II are computed after the simulation, in which t_{step} is the length of prediction step, N_{eval} is the number of scenarios to be evaluated. Three methods are compared in Table II: the original method, the constant-granularity with larger sample steps and the variable-granularity method. The experiment using the standard velocity profile (Figure 3) shows that the proposed variable-granularity prediction with $t_{sample} = [0.3, 0.9, 1.8, 3.0, 6.0]s$ has the lowest computational expense without sacrificing the overall performance. After porting the MATLAB code of the algorithm to C++ on the actual vehicle control platform, the MDP can plan for the strategy in around 1.5ms. Therefore, the algorithm can be run in real time on the autonomous vehicle at over 50Hz.

IV. PREDICTION OF LEADING VEHICLE BEHAVIORS

A. Extracting Leading Vehicle Acceleration

In autonomous driving, a better prediction of surrounding vehicles' behaviors will lead to more reasonable driver decisions and better overall system performance and robustness. As described in Section II-B, the motion model of the MDP makes predictions assuming that the leading vehicle will keep constant velocity, which is also the assumption of most Adaptive Cruise Control (ACC) algorithms. However, in some scenarios such as stop-and-go, the constant-velocity assumption is obviously not good enough and may lead to worse comfort and fuel efficiency. This prediction model does not take the acceleration of the leading vehicle into account in predicting its future velocity .

TABLE III
TOTAL COSTS OF USING DIFFERENT PREDICTION MODELS

Prediction Model	Parameter	Normalized Cost
$acc_{i=0,1,\cdots} = 0$		1.00
$acc_{i=0,1,\cdots} = acc_{est}$		1.48
$acc_{i=0,1,\cdots} = e^{-\lambda i} acc_{est}$	$\lambda = 0.10$	0.83
$acc_{i=0,1,\cdots} = e^{-\lambda i} acc_{est}$	$\lambda = 0.20$	0.80
$acc_{i=0,1,\cdots} = e^{-\lambda i} acc_{est}$	$\lambda = 0.30$	0.85

Most current Adaptive Cruise Control (ACC) systems depend on radar [1], which is able to sense the velocity of the lead vehicle. However, the range-rate will vary between multiple cars in dense traffic. In general, by fusing radar with other sensors such as LIDAR, as on our experimental platform, higher accuracy and reliability of the range and range-rate data and the estimated acceleration can be achieved. To compute the acceleration of the lead vehicle from the current perception system inputs of position and velocity, a 5-point discretized 1-D Derivative of Gaussian kernel is built, as shown in Equation 7, in which $\eta_1 = 0.153$, $\eta_2 = 0.347$, $\eta_3 = 0$, $\eta_4 = -0.347$, $\eta_5 = -0.153$.

$$acc_{est} = \eta_1 v_{t-4} + \eta_2 v_{t-3} + \eta_3 v_{t-2} + \eta_4 v_{t-1} + \eta_5 v_t \quad (7)$$

Then, the acceleration of the leading vehicle is used in the motion model for prediction.

B. Decaying Acceleration Prediction Model

As the behavior of the leading vehicle is uncertain, the MDP transition between states has many possibilities. But the probabilistic model of the leading vehicle's behavior is difficult to build by using only the information in the state vector X. Therefore, instead of modeling the probability of each potential velocity change of the leader, a statistically-based prediction model is proposed. It does prediction assuming that the acceleration of the vehicle will decay to zero. The model parameter is selected based on statistical data from the simulation using the standard velocity profile, and its performance is compared with other prediction models, as shown in Table III.. The result shows that with decaying acceleration prediction, an exponential decay parameter of $\lambda = 0.2$ leads to the best overall performance. Its total cost is 20% lower than using the constant-velocity assumption for prediction.

V. Point-based Markov Decision Process (QMDP) Behavior Control

A. Introduction of QMDP Behavior Control

The autonomous vehicle may experience considerable uncertainties while driving on the road. First, the behavior of the leading vehicle in the single-lane autonomous driving task is hard to predict. Second, significant sensor noise will affect decision making, as in the case of occasional sensor failure or extreme weather conditions. Third, the vehicle may not detect surrounding vehicles well due to the constraints of the sensing environment, such as obstacles blocking sensor observation. The effect of this uncertainty is compensated by using a statistical decaying acceleration model described in Section IV. Therefore, extending the MDP reasoning framework to QMDP, which can take the other two kinds of uncertainties into account, may improve the robustness of the autonomous system.

Based on the MDP controller, a Point-based MDP (QMDP) is built. In QDMP, Monte Carlo sampling is introduced in prediction and evaluation. Instead of computing cost by assuming the input state is deterministic and accurate, the total cost of $N_{samples}$ particles sampled from the distribution of the sensor input is computed using Equation 8,

$$C_{strategy} = \sum_{p=0}^{N_{samples}} \eta Prob(X_p) C_{X_p,strategy} \qquad (8)$$

in which Cstrategy is the final cost for the strategy, X_p is the sampled initial state, $Prob(X_p)$ is the probability X_p is the actual state, η normalizes the probability, $N_{samples}$ is the total number of samples and $C_{X_p,strategy}$ is the strategy cost given an initial state X_p. This integral cost can better represent the effect of the uncertainty to the strategy selection.

B. Sensor noise

The first kind of perception uncertainty is the sensor noise. As there is a Kalman filter in the perception system, the input from it is usually modeled as a normal distribution. It is difficult to sample many points to represent the Gaussian distribution due to the strict real-time requirement of the autonomous pilots decision making. However, as the noise of those sensors is already modeled as a Gaussian distribution, the number of samples can be efficiently reduced by using several fixed points corresponding to the Gaussian distribution parameters. In the experiment, three points $-\lambda, 0, +\lambda$ are selected from the Gaussian distribution and the costs corresponding to these sample points are weighted, corresponding to the probability density at those points.

The QMDP-based algorithm's ability to deal with sensor noise was tested in simulation. In the first test, the vehicle's velocity sensing was noise-free, and the estimate of the leading vehicle's acceleration was found by differencing the velocity and adding Gaussian noise. The result is shown in Figure 5. In the second case, the velocity of the leading vehicle has Gaussian noise added to it with standard deviations $\lambda = 0.0, 1.0 and 3.0 m/s$. Though in this case the acceleration found by differencing becomes too noisy to be usable in prediction and is therefore assumed to be zero, the algorithm still performs well. The result is shown in Figure 6. Both tests show that the QMDP-based algorithm performs well with different driving conservativeness at different uncertainty levels. The larger the uncertainty, the more conservative the autonomous control. This is because the cost function at such a sample point usually has a much higher safety cost than the other two samples and results in a higher weighted average cost compared to only considering the mean values of the input.

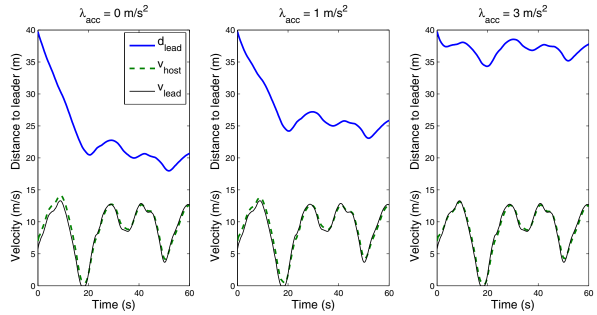

Fig. 5. Reasoning under acceleration noise

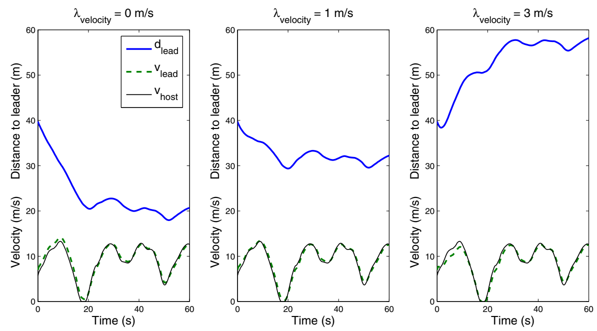

Fig. 6. Reasoning under velocity noise

TABLE IV

Total costs with and without reasoning under uncertainty

λ_{noise}	$C_{uncertainty}/C_{original}$
0.0	1.00
1.5	0.80
3.0	0.44

Table IV shows the performance under noisy perception input, in which the velocity noise at each time step is generated from a Gaussian distribution with $\mu = 0, \lambda = 0.0, 1.5, 3.0 m/s$. In Table IV, $C_{original}$ is the cost of the original algorithm assuming the perception is accurate, and $C_{uncertainty}$ is the cost of reasoning under sensor input uncertainty. Given the presence of sensor noise, the point-based algorithm reduces safety cost by 13% to 21%, which means the autonomous vehicle effectively avoids unsafe situations with high safety cost, while following a leading vehicle with the standard velocity profile.

C. Perception constraints

The other kind of uncertainty to be considered in autonomous driving systems is the sensing range constraints.

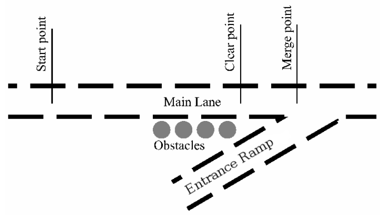

Fig. 7. Close-to-entrance scenario in single lane autonomy

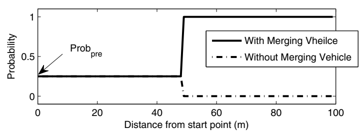

Fig. 8. Probability model in close-to-entering ramp scenarios

Though some widely used sensors can achieve a maximum sensing range of 100-200 meters theoretically, sensors are often blocked by obstacles in the environment. An example is urban intersections with narrow streets and close-in buildings, conditions under which human drivers usually exercise greater care and may slow down to better react to unexpected vehicles or pedestrians. This kind of uncertainty cannot be easily detected using only the perception system input.

The ability to perform conservatively in potentially unsafe situations is also important for single-lane autonomous freeway driving. An example is a scenario in which an autonomous vehicle is approaching an entrance ramp where it might be challenging for on-car sensors to detect merging vehicles, as shown in Figure 7.

To address the above, a probability model is built based on human domain knowledge of the entrance ramp traffic, as shown in Figure 8, in which $Prob_{pre}$ is the a priori probability that there will be a merging vehicle at the entrance, start point is the point where the probabilistic-based reasoning begins to be used, clear point is the position where the vehicle can get an unobstructed observation, and merge point is where the merging vehicle is supposed to appear in the autonomous vehicles lane. When the autonomous vehicle is approaching the start point, it begins to consider the probability of a vehicle entering from the ramp. Before it reaches the clear point, if it detects a merging vehicle, it will then reason using the deterministic perception input. If the perception system detects nothing before reaching the clear point, the behavior layer assumes the merging vehicle may appear with some probability. It will conservatively predict that one vehicle may appear at the merge point at a lower speed with a pre-defined probability. If there is still no car observed when the vehicle reaches the clear spot, the deterministic reasoning algorithm is then used.

In the simulation to verify this probabilistic reasoning mechanism, we assume that the autonomous vehicle will not get any information about the potential merging vehicle until it reaches the clear point. The performance with and without probabilistic reasoning (PR) is shown in Figure 9,

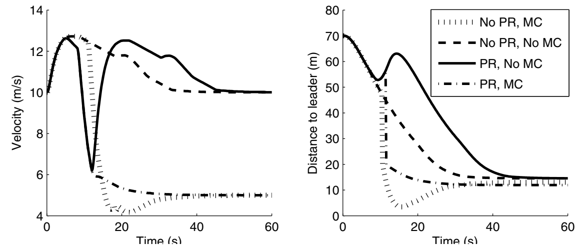

Fig. 9. Performance with and without Probabilistic Reasoning (PR)

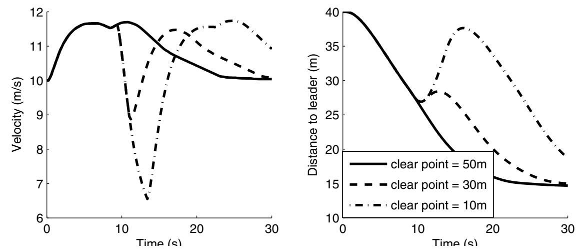

Fig. 10. Performance with different clear point positions

Fig. 11. Map of test road

in which MC is the merging car. The probabilistic reasoning algorithm will assume the merging vehicles will appear with 50% probability ($Prob_{pre} = 0.5$) with speed $5m/s$.

Figure 9 shows that with PR, the vehicle can slow down to avoid potentially merging vehicles. Without PR, the vehicle may get too close to the unexpectedly merging-in cars.

Results using different clear distances are shown in Figure 10. Figure 10 shows that when the autonomous vehicle is able to observe obstacles far away, there is no need for it to slow down significantly to avoid potential danger. When the merging vehicles at the entrance ramp cannot be detected, the autonomous vehicle slows down and conservatively passes the entrance ramp.

D. Road Test

To verify the QMDP-based algorithm's performance, two separate road tests were implemented using the autonomous vehicle Boss from the Urban Challenge 2007. The test field is a 1000-meters long single-lane road with an entrance ramp at the 300-meter point, as shown in Figure 11.

In the first test, the sensor noise's effect on the autonomous vehicle's behavior was evaluated. The test was implemented in a stop-and-go scenario using the vehicle's perception system and a simulated leading vehicle velocity uncertainty. The vehicle's distance keeping behavior is shown in Figure 12.

When the uncertainty was low ($\lambda = 1m/s$), the vehicle's behavior was not obviously affected, since the host vehicle was in most cases still in a relatively safe region even when the estimated leading vehicle velocity was faster than the

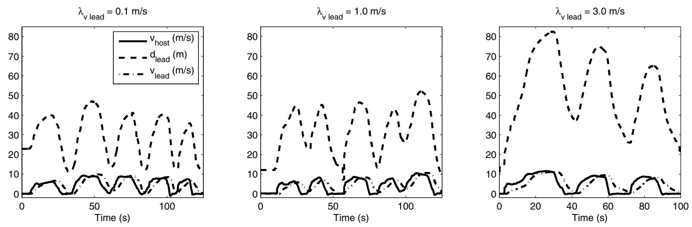

Fig. 12. Road test result under observed velocity noises of (left to right) $0.1m/s$, $1.0m/s$, $3.0m/s$

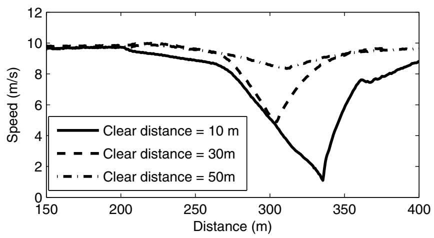

Fig. 13. Road test in close to entrance ramps scenario under sensing constraints

actual value. When the uncertainty was high ($\lambda = 3m/s$), the vehicle kept a larger distance to the leader to avoid potentially dangerous situations.

The second road test focused on verifying the QMDP-based autonomous driver's ability to deal with entrance ramps with static obstacles blocking the view of the vehicle sensors. The test field was set up with simulated clear distances of 10m, 30m and 50m. In the test, there were no entering vehicles at the entrance ramp. The test result is shown in Figure 13. The entrance ramp is at 350m. The host vehicle starts from a stop at 0m. The probabilistic reasoning begins at the 200m point to consider the probability of a merging vehicle. When the clear distance was small, the vehicle approached the entrance ramp or intersection more conservatively. When the clear distance was larger, the vehicle slightly adjusted its velocity and returned to normal speed to pass the entrance ramp once its perception system found that the intersection was clear.

VI. Conclusion and Future Work

In summary, in this paper, the point-based Markov Decision Process (QMDP) is used to improve the robustness of single-lane autonomous driving. Three kinds of uncertainties, surrounding vehicles' behaviors, sensor inputs and sensing ability constraints, are handled by the proposed algorithm, and its performance is verified in both simulation and in road tests. Compared with prior state-of-the-art Adaptive Cruise Control (ACC) and autonomous driving algorithms, the QMDP autonomous driving algorithm achieves better avoidance of unsafe behaviors caused by these three kinds of uncertainties, while maintaining satisfactory robustness and single-lane autonomous driving performance.

As future work, to achieve a better performance robustness, the parameters in the probability model such as the Probpre and the position of the clear point need to be refined based on both statistical and real-time data of the traffic. Second, to avoid potential unsafe situations, the vehicle performs conservatively in scenarios with higher uncertainty using the proposed algorithm. Though this behavior is reasonable in most cases, the influence of the host vehicle's conservative behavior on surrounding vehicles is not fully modeled in this framework. For instance, if the autonomous vehicle slows down close to an entrance, the vehicle that originally planned to yield to the autonomous vehicle may misunderstand the behavior and decide to merge in front of the autonomous vehicle. This social behavior may also need to be modeled for further development [15], [16]. Finally, further testing under different scenarios (speed and acceleration variations, etc.) will provide a more thorough evaluation of the proposed algorithm.

Acknowledgment

This work was supported by General Motors through the GM-Carnegie Mellon Autonomous Driving Collaborative Research Laboratory. The authors also would like to thank the Tartan Racing Team in the DARPA Urban Challenge, who built the vehicle control platform and the simulator.

References

[1] A. E. Ardalan Vahidi, "Research advances in intelligent collision avoidance and adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 143–153, 2003.

[2] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] C. Baker and J. Dolan, "Traffic interaction in the Urban Challenge: Putting boss on its best behavior," in *International Conference on Intelligent Robots and Systems (IROS 2008)*, 2008, pp. 1752–1758.

[4] J. Wei and J. M. Dolan, "A robust autonomous freeway driving algorithm," *2009 IEEE Intelligent Vehicle Symposium*, 2009.

[5] J. Wei, J. M. Dolan, and B. Litkouhi, "A prediction- and cost function-based algorithm for robust autonomous freeway driving," *2010 IEEE Intelligent Vehicle Symposium*, 2010.

[6] ——, "A learning-based autonomous driver: Emulate human drivers intelligence in low-speed car following," *SPIE Defense, Security, and Sensing 2010*, 2010.

[7] A. Hegyi, "Model predictive control for integrating traffic control measures," *TRAIL Thesis Series T*, vol. 2004, 2004.

[8] R. Sukthankar *et al.*, "A simulation and design system for tactical driving algorithms," *Proceedings of AI, Simulation and Planning in High Autonomy Systems*, 1996.

[9] D. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," 1989.

[10] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, p. 57, 2002.

[11] K. Murphy, "A survey of POMDP solution techniques," *environment*, vol. 2, p. X3, 2000.

[12] M. Spaan and N. Vlassis, "A point-based POMDP algorithm for robot planning," in *IEEE International Conference on Robotics and Automation*, vol. 3. Citeseer, 2004, pp. 2399–2404.

[13] M. Barth, F. An, T. Younglove, G. Scora, C. Levine, M. Ross, and T. Wenzel, "Comprehensive Modal Emission Model (CMEM), Version 2.0 Users Guide," *University of California, Riverside*, 2000.

[14] E. Camacho and C. Bordons, *Model predictive control.* Springer Verlag, 2004.

[15] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[16] F. Broz, "Planning for human-robot interaction:representing time and human intention," Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 2008.