A Real-Time and Predictive Trajectory-Generation Motion Planner for Autonomous Ground Vehicles

Junxiang Li, Bin Dai, Xiaohui Li, Chao Li
College of Mechatronic Engineering and Automation
National University of Defense Technology
Changsha, Hunan, 410073, P.R.China
Email: junxiangli90@gmail.com; bindai.cs@gmail.com;
xiaohui_lee@outlook.com;lichao.bit@foxmail.com

Yi Di
School of Mechanical Engineering
Nanjing University of Science and Technology
Nanjing, Jiangsu, 210094, P.R.China
Email: diyi8710@163.com

***Abstract*—This paper presents a real-time and predictive motion planner for autonomous ground vehicles. The motion planner can generate kinematically feasible and human-driving like trajectories based on an improved state-space trajectory generation method. Meanwhile, the motion planner also considers the future behavior of other participant vehicles through a control-space based Kalman predictor. The experimental results demonstrate that proposed motion planner has an improvement in generating safer and smoother trajectories, compared with Integrated Local Trajectory Planning (ILTP). Our motion planner has the capability to deal with complex traffic environments, especially with interactions of other participant vehicles.**

***Keywords*-motion planning; autonomous ground vehicles; trajectory-generation based approach; Kalman predictor.**

I. Introduction

In the past decades, the development of autonomous ground vehicles (AGVs) has been drawing great attention. Motion planning is a core technology for AGVs. A motion planner is responsible for generating safe and feasible trajectories considering both interactions with other vehicles and obstacles avoidance. Generally, developing a reliable and robust on-road motion planner faces following challenges: 1) how to generate kinematically feasible trajectories in real time with limited on-board computational resources; 2) how to deal with real-time dynamic changes in surrounding environments with the prediction of other vehicles' movements.

In recent years, a significant number of new techniques have been proposed to develop motion planners. The current mainstream techniques can be roughly categorized into two approaches: the graph-search based approach and the trajectory-generation based approach. The graph-search based approach (e.g., in [1]–[3]) is suitable for AGVs navigating in complex unknown environments only at low speeds. Also, most of the methods based on graph-search generate paths comprised of short-term precomputed path primitives without considering velocity planning. As a result, they will more easily cause stop-and-redirect motions. The trajectory-generation based approach is proposed to consider the vehicle kinodynamics in the motion planning. That is to say, velocity, acceleration, and force bounds are considered along with other constraints such as avoiding obstacles.

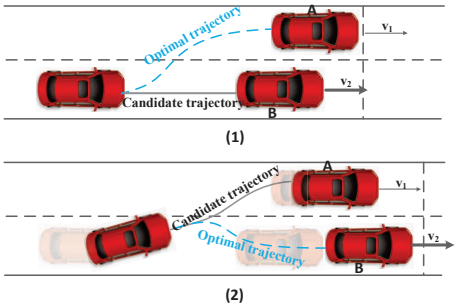

Figure 1. An example of motion planner without considering uncertainty.

The trajectory-generation based approach can be classified into two classes [4]: control-space trajectory-generation and state-space trajectory-generation. The trajectories generated in the control space (e.g., in [5], [6]) can satisfy vehicle kinematic constrains, but they may not satisfy environmental constraints such as road geometries. Conversely, the trajectories generated in the state space can easily guarantee the satisfaction of environmental constraints, though the generation process requires more computing resource than the one using control-space trajectory-generation approach. Thus, the state-space trajectory-generation approach is more widely used in many real AGVs (e.g., in [7], [8]). Our previous work in [9], [10] designs an integrated local trajectory planning and tracking control (ILTPTC) framework for AGVs, which can bridge the gap between motion planning and low-level controlling. It consists of a motion planner, which is also based on state-space trajectory-generation approach, combining with a trajectory tracking controller.

Most of previous work treats other traffic participants as static obstacles in each planning cycle and then chooses the optimal trajectory to avoid collision with all obstacles (including traffic participants and real static obstacles). For instance, as shown in Fig. 1. (1), the perception system of

978-1-5386-3022-8/17 $31.00 © 2017 IEEE
DOI 10.1109/IHMSC.2017.140

the AGV detects the distances to other traffic participants. Apparently, the motion planner chooses the safer trajectory shown as the blue dotted line because Vehicle A is farer to the AGV. However, after several consecutive planning cycles (shown in Fig. 1. (2)), the AGV will turn back to the former lane in that the velocity of Vehicle A is slower than Vehicle B. As a result, the motion of the AGV does not behave as driven by a human [11] even if the AVG can drive autonomously.

Consequently, the behavior prediction of dynamic vehicles in the motion planning is essential. This paper presents a real-time motion planner with consideration of the movements of other participant vehicles. The motion planner of our AGV is based on the state-space trajectory-generation method, because the method easily satisfies constraints of road geometries. Then, a control-space based Kalman predictor which can predict the behavior of other participant vehicles is designed. The predictor based on control-space rather than state-space [12] because it is effective and requires less computing resource.

The main contributions of this paper include two aspects. Firstly, the proposed motion planner can generate kinematically feasible and human-driving like trajectories. The planned trajectories are safer and smoother than that of ILTPTC [9]. Secondly, the proposed motion planner can predict the movements of other participant vehicles in complex situations in real time.

The remainder of the paper is organized as follows. In Sec. II, the system framework as well as the relationship between our predictive motion planner and ILTPTC is described. In Sec. III, more specific method of our real-time and predictive motion planner is discussed. Sec. IV presents the simulation experiments and results. Finally, Sec. V draws conclusions and suggests future work.

II. System Framework

The framework of our AGV software system is shown in Fig. 2. ILTPTC (integrated local trajectory planning and tracking control) [9] we proposed previously is an integrated framework that contains a motion planner, ILTP (integrated local trajectory planning), and a trajectory tracking controller. This paper proposes a new predictive motion planner KPMP (Kalman Predictor-based Motion Planner) to replace ILTP in the ILTPTC framework. Specifically, KPMP has four components in total: 1) a path planner; 2) a velocity planner; 3) a participant vehicles predictor; and 4) an optimization performance evaluator. KPMP upgrades ILTP and improves the performance of motion planning in generating safer and smoother trajectories.

III. Method

This section introduces the details of our KPMP. According to the sequence of a planning cycle, our KPMP can be

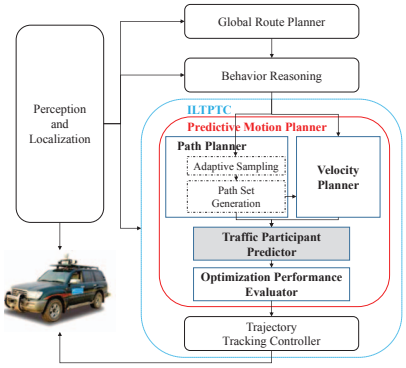

Figure 2. System framework.

divided into four parts: 1) path planning; 2) velocity planning; 3) prediction of participant vehicles; and 4) collision-test and performance evaluation.

A. Path Planning

Path planning generates feasible paths which can be easily followed by the low-level controller as well as comply to the environmental constrains. Thus we develop our path planning algorithm based on the state-space trajectory-generation method.

Firstly, we apply the adaptive sampling algorithm [4] to sample the terminal states (i.e., position coordinates, headings, curvatures) adaptively based on the road geometry. Then, a kinematically feasible path set from the initial state is generated based on the model-predictive generation method [13]. We exploit the vehicle kinematic model as the predictive motion model. The constrains can be written as $X_f = f(X_i, u)$, where X_i is the initial state, X_f is the terminal state. The function $f()$ is the vehicle kinematic model, which can be represented by the following equations:

$$\begin{aligned} \dot{x} &= v * cos(\theta) \\ \dot{y} &= v * sin(\theta) \\ \dot{\theta} &= \kappa * v \\ \dot{\kappa} &= u \end{aligned} \tag{1}$$

where (x, y) is position coordinate, θ is heading, κ is curvature, v is vehicle velocity, and u is steering control.

As we know, continuous curvature can guarantee the smoothness of a path and avoid jerky steering actions. Thus, the curvature is represented as a polynomial function of the arc length to ensure the continuity of each path's curvature. In order to obtain a unique solution, the number of unknown parameters in $\kappa(s)$ must be equal to the number of useful known conditions. Note that the initial curvature $\kappa(0)$ determines the zero-order parameter $a = \kappa(0)$. Other useful conditions left are the terminal state and the initial position. Thus, a cubic polynomial function of variable s is adopted. The unknown parameters involve $q = [b, c, d, s_f]$.

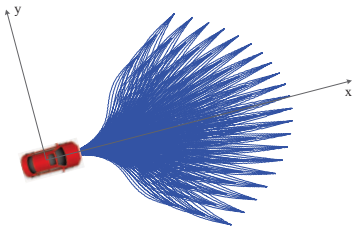

Figure 3. Visualized lookup table storing initial guesses for unknown parameters.

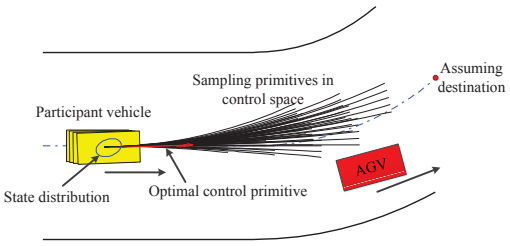

Figure 4. The control-space based Kalman predictor.

Since it is nontrivial to obtain an analytical solution for such differential equations of strong nonlinearities, we use Newton's method to solve equations numerically. Specifically, the unknown parameters are iteratively calculated through finding the minimal value of the error ($\Delta X_f(q)$).

As a nonlinear optimization problem, an initial guess for the unknown parameter q_0 can greatly affect the convergence of iterations. Hence, inspired by the work of [14], we propose a precomputed lookup table storing initial guesses for unknown parameters. As illustrated in Fig. 3, we sample several final states, and then generate corresponding paths based on the vehicle body-center coordinate framework. The parameters of these generated paths are stored as the initial guesses.

B. Velocity Planning

This subsection discusses the velocity planning algorithm. Firstly, we calculated a maximal velocity limit for the desired velocity. Then, the desired velocity is obtained by a presented function. Lastly, a velocity profile that can be executed by the velocity controller (i.e., throttle/brake) is designed. Note that the velocity profile needs to be designed because our AGV has an integrated framework of motion planner and controller.

To ensure safety, we set a maximal velocity limit (V_{max}) which is determined by the following three aspects:

$$V_{max} = inf\{V_{mal}, V_{mlat}, V_{mlon}\} \tag{2}$$

where maximal allowed velocity (V_{mal}) is estimated by the high-level behavior reasoning module. Maximal lateral velocity (V_{mlat}) is constrained to prevent side skids of vehicles and related to the curvature of the road and the maximal lateral acceleration: $V_{mlat}^2|\kappa| \leq Acc_{mlat}$. Maximal longitudinal velocity (V_{mlon}) is limited by the following equation: $(V_{mlon}^2)/(2Dec_{mlon}) + D_{safe} \leq s_f$, where Dec_{mlon} is the maximal longitudinal deceleration; D_{safe} is the safe braking distance; and s_f is the length of the candidate paths.

Then, we design a univariate function (3) of the length of generated paths to calculate $V_{desire} \in [0, V_{max})$.

$$V_{desire} = \begin{cases} \frac{2V_{max}}{1+exp(D_{safe}-s)} - V_{max}, & \text{if } s \geq D_{safe} \\ 0, & \text{if } s < D_{safe}. \end{cases} \tag{3}$$

Furthermore, we divide the velocity profile into two segments. The first segment is a cubic polynomial function, which can ensure the continuous acceleration/deceleration. The second segment is a stable period of maintaining the desired velocity.

The initial velocity v_0 can be obtained. Setting the initial acceleration a_0 and final acceleration a_f to 0. The final velocity v_f, which equals to the desired velocity, can be obtained by (3). With the path length s_f and $s(0) = 0$, the unknown parameters $\{k_0, k_1, k_2, k_3\}$ in the polynomial function and the final time t_f can be analytically solved.

C. Prediction of Participant Vehicles

Inspired by the control-space trajectory-generation approach, which is based on the vehicle dynamics and requires less computational complexity than the state-space trajectory-generation approach, our control-space based Kalman predictor (shown in Fig. 4) is proposed.

The trajectory prediction of participant vehicles is determined by their current states and future control of the participant vehicles. The current states can be measured by the perception system of an AGV. However, the future control of each participant vehicle is unknown. Thus, for each participant vehicle, the possible control primitives can be sampled in its control space. Then, the sampled control primitives are evaluated, and the optimal one is chosen as the best estimation of this traffic participant's future control. Once obtaining the future control and the current state, the future trajectory and the distribution of the state along this trajectory can be computed using a Kalman predictor. The whole process is repeated for all participant vehicles in the horizon of the AGV. For simplicity, only one participant vehicle is taken as an example.

In order to simplify the calculation, given the discretization time step Δt, the kinematic model of participant vehi-

cles can be generally linearized as (4):

$$\begin{aligned}
&\hat{X}_{t+1} = A\hat{X}_t + B\hat{u}_{t+1},\\
&A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},\\
&B = \begin{bmatrix} cos(\theta_t)\Delta t & 0 \\ sin(\theta_t)\Delta t & 0 \\ 0 & \Delta t \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.
\end{aligned} \tag{4}$$

where the state of the participant vehicle at stage t is denoted as $\hat{X}_t = [x_t, y_t, \theta_t, v_t, w_t]'$; (x, y) is the position coordinate; θ is the heading; v is the linear velocity; w is the angular velocity; and the control primitive at stage t is denoted as $\hat{u}_{t+1} = [v_{t+1}, w_{t+1}]'$.

At first, the possible control primitives of a participant vehicle are sampled in the control space U_s. The sampling interval ($[v_{min}, v_{max}], [w_{win}, w_{max}]$) and the sampling step can be tuned according to the environments. Then, the optimal control primitive is chosen by a cost function, which considers both the collision avoidance with other obstacles (expect itself and the AGV) and the distance to an assumed destination. There are two cases of the assumed destination depending on the driving direction of the participant vehicle. If the participant vehicle has the same driving direction with the AGV, the destination is the far point (in the horizon of the AGV) on the reference line. Otherwise, the destination is the position of the AGV, that is to say, the participant vehicle aims to drive towards the AGV. We make the latter assumption because the safety of the AGV can be maximized if the AGV can avoid the unreasonable behavior. Meanwhile, it is also assumed that when predicting the motion for a traffic participant, all the other vehicles (including the AGV) will keep moving with fixed velocities.

Once choosing the optimal control primitive ($\hat{u}^*_{t+1}$) of a participant vehicle as the future control, the state equation and the observation equation of the participant vehicle can be linearized as follows:

$$\hat{X}_{t+1} = A\hat{X}_t + B\hat{u}^*_{t+1} + q_t, \quad q_t \sim N(0, Q) \tag{5}$$

$$\begin{aligned}
&Y_t = C\hat{X}_t + r_t, \qquad r_t \sim N(0, R)\\
&C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}.
\end{aligned} \tag{6}$$

The distributions of noises (q, r) are assumed to be zero-mean Gaussian, with covariance matrix Q and R respectively. Then, a Kalman predictor is exploited to predict the future state at stage $t + 1$ as follows:

$$\begin{aligned}
&\hat{X}_{pre} = A\hat{X}_t + \hat{u}^*_{t+1},\\
&P_{pre} = AP_tA^T + Q,\\
&K_t = P_{pre}C^T(CP_{pre}C^T + R)^{-1},\\
&\hat{X}_{t+1} = \hat{X}_{pre} + K_t(Y_t - C\hat{X}_{pre}),\\
&P_{t+1} = (I - K_tC)P_{pre},
\end{aligned} \tag{7}$$

where K_t is the Kalman gain; $\hat{X}_{pre}$ is the priori state estimate; $\hat{X}_{t+1}$ is the posteriori state prediction; Y_t is the partially observable state; P_{pre} is the priori covariance matrix of the state; P_{t+1} is the posteriori covariance matrix of state, which reflects the uncertainty of the prediction.

As we know, given an upper bound of the probability θ, the mean value $\hat{X}_{t+1}$, and the covariance matrix P_{t+1} , an error ellipse (shown in Fig. 4) can be displayed as the distribution of the future state along the trajectory (i.e., generated by the optimal control primitive). It also provides an upper bound of the collision probability θ for the participant vehicle if there is an intersection between the ellipse and the obstacles. This ellipse will be used in the collision-test for the participant vehicles in the Sec. III-D.

D. Collision-Test and Performance Evaluation

This section discusses the collision-test and the performance evaluation. The candidate trajectories will be trimmed if they collide with obstacles (in this paper, only participant vehicles are considered). The trimming method is discussed in [9], [10]. As the motion of the participant vehicles have been predicted in Sec. III-C, a participant vehicle is approximated by a bigger ellipse, which is the Minkowski sum of the circle and the error ellipse. Note that the circle is chosen from the circle sets based on the driving direction. If the extracted point intersects the bigger ellipse, the trajectory will be trimmed likewise.

After the collision-test, the performance of each available trajectory is evaluated by a function (8). The optimal trajectory is the one that minimizes the total cost. All of cost terms are normalized to be within [0,1]. N points are extracted from each candidate trajectory for the cost calculation.

$$i^* = \arg\min_{i=1}^{M}(w_oC_o + w_dC_d + w_sC_s + w_vC_v) \tag{8}$$

where C_o, C_d, C_s, C_v are cost terms([9], [10]), which are designed as peferences for choosing trajectories; w_o, w_d, w_s, w_v are weight terms, which can be adjusted in practical according to the environments; i is the index of candidate trajectories; and M is the total number.

IV. Experiments and Results

In this section, the comparisons of planned trajectories using proposed KPMP and ILTP in different scenarios are demonstrated. Results were presented through simulation experiments. As the interference from the controller of the

AGV is expected to be minimum, the controller is assumed to track generated trajectories precisely.

A. Setup

We designed four comparative experiments to verify the performance of our motion planner KPMP, which contain: 1) driving without interactions on a straight road; 2) driving on a unidirectional two-lane road when there are vehicles with different velocities in front of the AGV; 3) driving on a unidirectional two-lane road when a vehicle is overtaking the AGV; 4) driving on a bidirectional two-lane road when there are vehicles with different driving directions.

Each experiment was repeated twenty times to verify the effectiveness of KPMP in dealing with different Gaussian noises of other participant vehicles. The experiments were simulated in Matlab R2012b, which runs on a PC (Intel(R) Core(TM) i5@2.20GHz, 4.0GB memory). The main parameters of the experiments are listed in Table I.

Table I
MAIN PARAMETERS IN THE SIMULATION ENVIRONMENT

Meaning	Value
Iteration Cycle	500ms
Lane Width	4 m
Vehicle Width	2 m
Vehicle Length	4 m
Parameter (W) in Performance Function	[0.6,0.1,0.1,0.2]

B. Results

The results of the experiments demonstrate that our KPMP has the same effect as ILTP when there is no dynamic obstacle. However, when the AGV interacts with participant vehicles, the safety and smoothness of trajectories generated by KPMP will be greatly improved. In all figures, the red curves are the planned trajectories in our simulation. The blue arrows represent the driving direction of vehicles at their initial positions. Important planning cycles are denoted as the index added a symbol # ahead (e.g., the 6th planning cycle is denoted as #6 in figures).

Firstly, we made a comparison of two approaches under the circumstance that the AGV drives without interactions on a straight road. The results indicated that KPMP was verified to have the same capability as ILTP in dealing with the environments without interactions. As shown in Fig. 5, to avoid the static obstacles (grey ones), the autonomous vehicles (red ones) select a longer, but still smooth path instead of doing short-cuts. The velocity decreases when lane changing in each approach.

Secondly, simulations were performed on a unidirectional two-lane road, with vehicles of different velocities in front of the AGV. These tests were implemented to validate that KPMP generates more reasonable trajectories than ILTP in this scenario. As shown in Fig. 6, the velocities of two

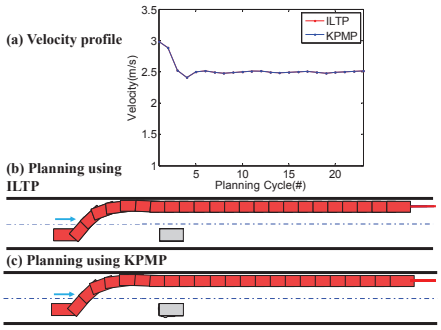

Figure 5. Driving without interactions on a straight road.

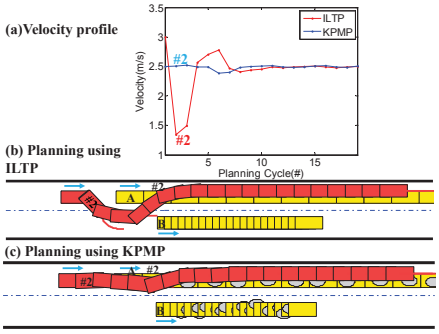

Figure 6. Driving with vehicles in front of the AGV on a unidirectional two-lane road.

vehicles are respectively 4m/s (Vehicle A), 1m/s (Vehicle B). KPMP does not choose an overtake action at #2 because it can predict that the participant Vehicle A in the same lane would drive quickly out of horizon. As a consequence, the trajectories generated by KPMP is smoother and more human-driving like than ILTP because the vehicle does not change its lane frequently. It is obvious that velocity profile of the AGV using KPMP is also smoother. The AGV using ILTP needs to change lanes in a very short time so that there is a sharp decrease of velocity at the beginning.

Thirdly, simulations were also performed on a unidirectional two-lane road, but with one vehicle overtaking the AGV. As shown in Fig. 6, the velocity of the front vehicle (Vehicle A) of the AGV is 2m/s, whereas the velocity of the overtaking vehicle (Vehicle B) in the next lane is 5.5m/s. ILTP cannot predict the Vehicle B is overtaking, thus making the trajectories dangerous (e.g., #2 and #3) and jerky. In contrary, KPMP predicts the motion of Vehicle B, thus planned trajectories avoid the collision. The AGV plans to change lanes only after Vehicle B have overtaken and gone away (e.g., #6).

Finally, we emulated driving with interactions on a bidirectional two-lane road. As Fig. 8 illustrates, an AGV and an oncoming vehicle (Vehicle A) drive on the same lane, meanwhile a vehicle (Vehicle B) drives with the same driving direction of the AGV on the other lane. The velocity

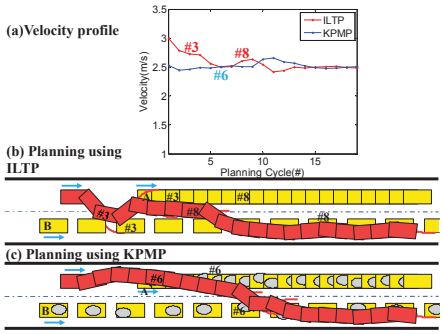

Figure 7. Driving with vehicles overtaking the AGV.

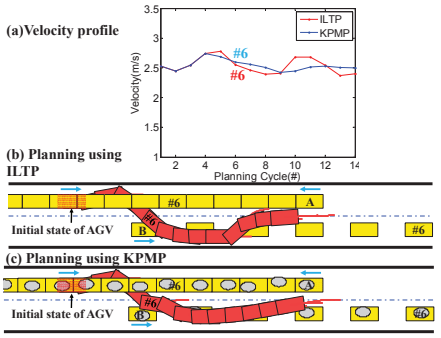

Figure 8. Driving with vehicles on a bidirectional two-lane road.

of Vehicle A is 4m/s, whereas the velocity of Vehicle B is 8m/s. Because the AGV is near to Vehicle B at the initial state, two motion planners both choose a longer trajectory to avoid the collision rather than changing lanes at first. However, after #6, predicting the position and the driving direction of Vehicle A, KPMP chooses a smoother trajectories than ILTP. The AGV tends to drive along the reference road, which is reasonable. The desired velocities of KPMP are more stable than that of ILTP.

V. Conclusions

The presented motion planner KPMP has generated safe and smooth trajectories in real time by considering the future behavior of other participant vehicles. The trajectories generated by KPMP are not only kinematically feasible, but also environmental adaptive. An original and effective control-space based Kalman predictor can make the AGV interactively respond to real-time changes in a realistic environment.

The results show that the safety and smoothness of trajectories generated by KPMP will be greatly improved as to previous motion planner ILTP, especially when the AGV interacts with dynamic vehicles. For the future work, we will design a more robust motion planner with considering the motion prediction of moving pedestrians and the uncertainty of an AGV.

Acknowledgment

This work was supported by the China Scholarship Council (CSC) and the National Nature Science Foundation of China under No.61375050 and No.91220301.

References

[1] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[2] M. Du, T. Mei, H. Liang, J. Chen, R. Huang, and P. Zhao, "Drivers' visual behavior-guided RRT motion planner for autonomous on-road driving," *Sensors (Switzerland)*, vol. 16, no. 1, pp. 1–19, 2016.

[3] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1961–1976, 2015.

[4] T. M. Howard, C. J. Green, A. Kelly, and D. Ferguson, "State space sampling of feasible motions for highperformance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, no. 67, pp. 325–345, 2008.

[5] A. Cherubini, F. Spindler, and F. Chaumette, "A new tentacles-based technique for avoiding obstacles during visual navigation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4850–4855.

[6] S. M. Erlien, S. Fujita, and J. C. Gerdes, "Shared steering control using safe envelopes for obstacle avoidance and vehicle stability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 441–451, 2016.

[7] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[8] T. Gu, J. M. Dolan, and J.-W. Lee, "On-Road Trajectory Planning for General Autonomous Driving with Enhanced Tunability," in *Intelligent Autonomous Systems 13*. Springer, 2014, pp. 247–261.

[9] X. Li, Z. Sun, D. Cao, D. Liu, and H. He, "Development of a new integrated local trajectory planning and tracking control framework for autonomous ground vehicles," *Mechanical Systems and Signal Processing*, vol. 87, Part B, pp. 118–137, 2015.

[10] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 2, pp. 740–753, 2016.

[11] J. Mainprice, E. A. Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, "Planning human-aware motions using a sampling-based costmap planner," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5012–5017.

[12] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion Planning under Uncertainty for On-Road Autonomous Driving," in *2014 IEEE International Conference on Robotics & Automation(ICRA)*, 2014, pp. 2507–2512.

[13] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *The International Journal of Robotics Research*, vol. 26, no. 2, pp. 141–166, 2007.

[14] R. a. Knepper, S. S. Srinivasa, and M. T. Mason, "Toward a deeper understanding of motion alternatives via an equivalence relation on local paths," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 167–186, 2012.