Data-Driven State-Increment Statistical Model and Its Application in Autonomous Driving

Chao Ma, Jianru Xue, *Member, IEEE*, Yuehu Liu, *Member, IEEE*, Jing Yang, Yongqiang Li, and Nanning Zheng, *Fellow, IEEE*

***Abstract*— The aim of trajectory planning is to generate a feasible, collision-free trajectory to guide an autonomous vehicle from the initial state to the goal state safely. However, it is difficult to guarantee that the trajectory is feasible for the vehicle and the real path of the vehicle is collision-free when the vehicle follows the trajectory. In this paper, a state-increment statistical model (SISM) is proposed to describe the kinodynamic constraints of a vehicle by modeling the controller, the actuator, and the vehicle model jointly. The SISM consists of Gaussian distributions of lateral error increments in all state subspaces which are composed of the curvature radius, the velocity, and the lateral error. It is a data-driven modeling approach that can improve the SISM via increasing the number of samples of the increment-state, which is composed of the state and its corresponding increment of the lateral error. According to the SISM, the experience cost functions are designed to evaluate the trajectories for searching the best one with the lowest cost, and the real path can be predicted directly according to the planned trajectory and the vehicle state. The predicted path can be utilized effectually to evaluate the safety of the vehicle motion.**

***Index Terms*— Statistical model, data-driven, experience cost, trajectory evaluation, path prediction.**

I. Introduction

TRAJECTORY planning plays an important part in autonomous vehicle navigation. It is expected to generate a feasible and collision-free trajectory from an initial state to a goal state by taking both constraints of the scene and the motion of the vehicle into consideration. Generally, the feasibility of a trajectory means that it satisfies the nonholonomic constraints of the vehicle, and guarantees that the vehicle can follow the trajectory with a tolerable lateral error. The collision-free means that the vehicle can move safely in traffic scenarios when it follows the trajectory. Therefore, the objective of the collision-free, essentially, is the real path of the vehicle rather than the planned trajectory. As a consequence, it is necessary to evaluate the feasibility of the planned trajectory and predict the real path accurate enough to guarantee that the planned trajectory is safe for the vehicle.

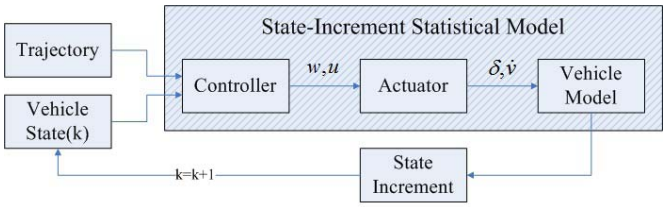


Fig. 1. The traditional vehicle model is used to describe the motion constraints of a vehicle. But the SISM models the integration of the controller, the actuator and the vehicle model. As a result, the SISM can be used to predict the state increment directly according to the planned trajectory and the current state of the vehicle. In the Figure, w is the steering wheel angle, u is the throttle pedal or brake pedal value, δ is the front wheel angle, and $\dot{v}$ is the longitudinal acceleration of the vehicle.

However, it is difficult to evaluate the trajectory feasibility and predict the real path effectively according to the traditional kinematic model of the vehicle due to two reasons. Firstly, the kinematic model cannot reflect the characteristics of the actuator, like the tire deformation, the delay and nonlinearity of the actuator response and so on. Secondly, the control algorithm decides directly whether the trajectory is feasible for the vehicle. But most trajectory planning methods ignore the constraints of control algorithm when they evaluate the feasibility of the planned trajectory.

In the 2007 DARPA Urban Chanllenge competition, most autonomous vehicles generated their trajectories according to the kinematic model of the vehicle [1]–[6]. But they ignored the actuator's constraints of different vehicles. To solve this problem, Kelly and Nagy proposed a reactive nonholonomic trajectory generation method via parametric optimal control by using cubic curvature polynomials in [7] and [8]. This method was utilized on the CMU's vehicle, Boss, which won the DARPA Urban Chanllenge competition [9], [10]. In 2010, Werling proposed a semi-reactive trajectory generation method by taking optimal-control strategies into consideration [11], [12]. These methods can generate a trajectory satisfying the constraints of the actuator, which means the trajectory following can be achieved via continuous control outputs.

However, control outputs to the actuator, in general, are calculated by the control algorithm according to the relationship between the vehicle and the trajectory. If the control algorithm cannot satisfy the need of the optimal control, the control outputs will not make the vehicle follow the

Manuscript received October 16, 2016; revised October 14, 2017; accepted December 26, 2017. This work is supported in part by National Key R&D Program Project 2016YFB1001004 and in part by NSFC Projects 61751308 and 61773311. The Associate Editor for this paper was M. Chowdhury. *(Corresponding author: Jianru Xue.)*

The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: machao0919@stu.xjtu.edu.cn; jrxue@mail.xjtu.edu.cn; liuyh@mail.xjtu.edu.cn; jasmine1976@mail.xjtu.edu.cn; keaijile321@stu.xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2797308

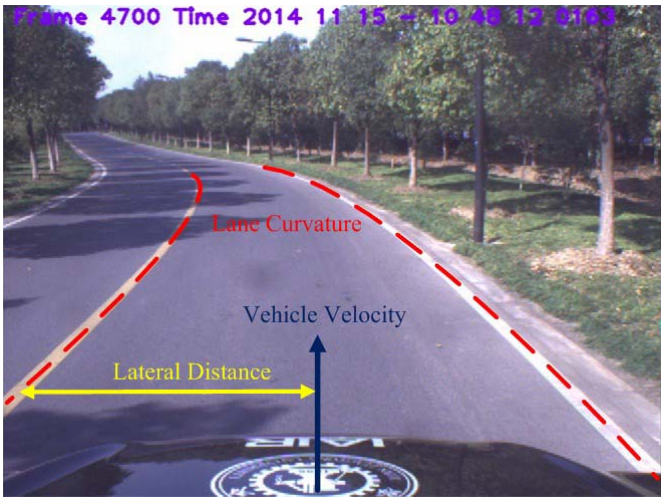


Fig. 2. The primary state factors of human driving experience: lane curvature, vehicle velocity and lateral distance.

trajectory accurately. Moreover, the actuator response also needs to be considered which may lead to possible risks of the vehicle. But it is difficult or expensive to consider jointly the constraints of the control algorithm, the actuator response and the kinematic constraints of the vehicle.

The aforementioned reasons lead to a serious problem for traditional trajectory planning methods which utilize the kinematic model of a vehicle. The planned trajectory will be unfeasible if the control accuracy is inadequate or some defects of the actuator exist, because the kinematic model cannot describe the constraints of the control algorithm and the actuator. To solve this problem, a data-driven statistical model is needed urgently, which can describe holistically the constraints of the controller, the actuator and the vehicle motion, as shown in Fig. 1.

In order to build a statistical model of a vehicle, we refer to learn from the human driver. When a human drives a vehicle, he/she does not care what the kinematic model is or how to model the actuator of the vehicle. All the driver does is to establish the mapping between the environment (the input) and the change of the vehicle state (the output). Through a long-term driving training, this mapping, namely the driving experience, will become more and more accurate and the driver can use it to predict the vehicle state in the future. In a simple traffic scene, when a driver drives on an empty road, he/she primarily cares about the curvature of the lane, the velocity of the vehicle and the lateral distance between the vehicle and the lane markings. After completing the instant driving operation, he/she will notice the change of the lateral distance chiefly to make sure that the vehicle can keep the lane, as shown in Fig. 2.

For the purpose of imitating the human driving experience, in this paper, we build a data-driven statistical model, the State-Increment Statistical Model (SISM), that regards the state, including the curvature radius of the trajectory, the vehicle velocity and the lateral error between the vehicle and the trajectory, as the inputs, and the lateral error increment as the output. Then we use the model to describe the mapping between the inputs and the output. More specifically, the SISM specifies a Gaussian distribution of lateral error increments for each discretized three-dimensional state subspace. Through analyzing lots of historical samples of the increment-state, which is composed of the state and its corresponding increment of the lateral error, we find that the distribution of lateral error increments can be approximated by a normal distribution in a small state subspace where the sample number is adequate. As a result, the mean and the standard deviation of lateral error increments can be obtained and used to predict the most probable real path of the vehicle. According to the mean and the standard deviation of increments, and the sample number of the increment-states in each state subspace, the experience cost can be calculated to evaluate the trajectories to ensure that the best trajectory is the most feasible one.

The rest of the paper is organized as follows. The collection approach of statistical samples is introduced in Section II. Section III describes the method to build the SISM. Section IV presents technical details of the motion prediction of the vehicle based on the SISM. The trajectory optimization method is described in Section V, and the experimental results are shown in Section VI.

A. Related Work

Many trajectory generation methods are applied in the autonomous vehicles, like RRT [1], [13]–[15], A^* [3], VFH [16], [17] and so on. The most of these methods generate trajectories by taking the kinematic constraints of the vehicle into account but ignoring the constraints of the control algorithm and the actuator.

In [7] and [8], Kelly and Nagy proposed a reactive nonholonomic trajectory generation method via parametric optimal control. This method can get optimal control inputs at any time theoretically to make the vehicle follow the trajectory accurately [18]. As a result, it can make sure that control inputs to the actuator are smooth and continuous. In addition, this solution gives a way to represent any trajectory by a curvature sequence. In [11] and [12], Werling *et al.* proposed a semi-reactive trajectory generation method in a Frenet Frame. This method generates a feasible trajectory by optimizing the lateral and longitudinal movements in a Frenet frame which is given by the tangential and normal vector at a certain point of reference curve. Similarly the relationship between the planned trajectory and the real path of the vehicle can also be represented in a Frenet frame where the tangential vector is the vehicle velocity and the normal vector is the lateral error.

However, the kinematic constraints of the vehicle in both aforementioned methods are also incomplete without taking the constraints of the control algorithm into account. Some approaches remedy the incomplete kinematic constraints by analyzing the human driving data. Because humans always treat the vehicle as a holistic object, namely build a holistic vehicle model subconsciously. As a result, these approaches take some decision parameters of the human driving as the output of the vehicle model according to some traffic scenarios. In [19], Yao *et al.* optimize the parameter of longitudinal length and lateral offset in lane change trajectory via the statistics of the human driving data. Gu and Dolan build a

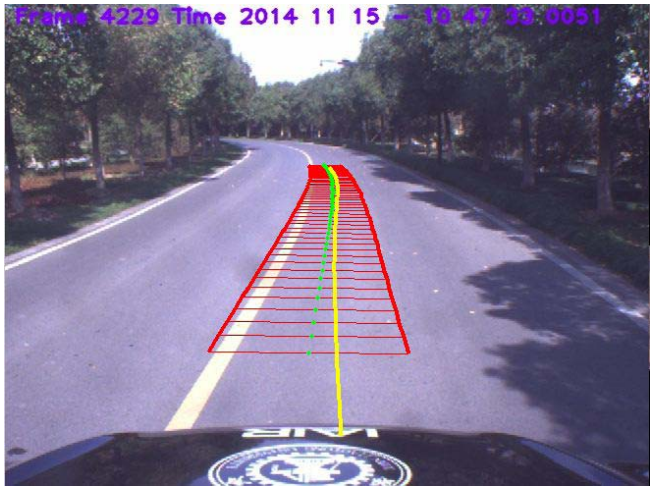


Fig. 3. The trajectory-band (the red area) collides with the lane marking, but its real path (the yellow line) is safe.

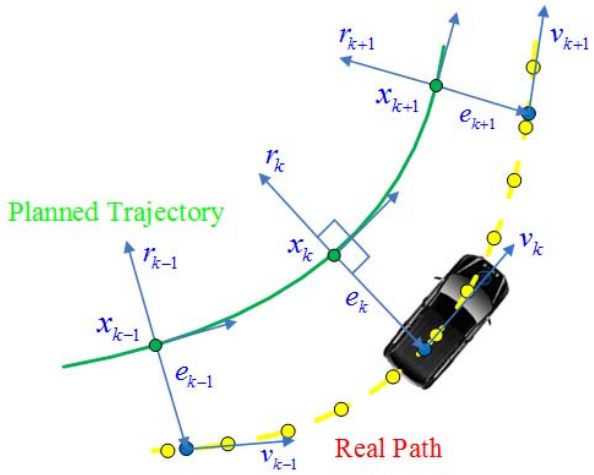


Fig. 4. The green points are the planned trajectory points, the yellow points are the collected real path points from GPS/IMU, and the blue points are the specific points on the real path corresponding to the planned trajectory points. Then the state $x_k = (r_k, v_k, e_k)$ and its corresponding increment-state $x_k^* = (x_k, \Delta e_k)$ are given, where r_k is the curvature radius of the planned trajectory point, v_k is the vehicle velocity, e_k is the lateral error, namely the distance between the planned trajectory point and its corresponding specific point. $\Delta e_k = e_{k+1} - e_k$ is the lateral error increment on the specific point.

parameterized speed model to fit human driving data in [20]. But these methods cannot be used to predict the real path of the vehicle directly.

Meanwhile, Kim and Yi build a holistic vehicle model by a maximum-likelihood filtering method to predict the vehicle motion probabilistically in [21]. But this model cannot describe the relationship between the vehicle and the planned trajectory. Therefore, it cannot be used for the vehicle navigation.

There are also some traditional statistical models to describe the vehicle state and predict the vehicle motion according to Mento Carlo methods [22]–[24]. These models need to establish the state equation to describe the state transition and add a Gaussian random noise to this state equation. Then the collected data are used to estimate the distribution of the noise. Finally the mean and the standard deviation of the Gaussian random noise can be obtained based on its distribution. However, the SISM is different from these statistical models. It doesn't need the state equation to describe the physical or mathematical transform relation of the states. Instead of the Gaussian random noise, the distribution of the lateral error increment is obtained directly from the collected data. This kind of data-driven modeling approach is closer to human cognitive approach than traditional statistical modeling approaches. Because humans do not care what the state equation is, but concern about whether the correlation of state parameters exists.

In our previous work [25], the State Statistical Model (SSM) is proposed to generate a trajectory-band which is able to predict the motion area of the vehicle. But there is a serious problem for the SSM, which is that the mean and the standard deviation of the lateral error are abnormally large because of the effect of the initial lateral error. Then the planned trajectory-band will be abnormally wide, because there is a positive correlation between the standard deviation of the lateral error and the width of the trajectory-band. As a consequence, it collides with the lane markings or obstacles sometimes, but the real path of the vehicle is safe, as shown in Fig. 3.

In order to solve this problem, we add a new parameter, the lateral error increment, to the SSM to build the State-Increment statistical Model (SISM) in this paper. As a result, the real path of the vehicle can be predicted by integrating lateral error increments. Compared with the SSM, the SISM is more accurate and the predicted path can reflect the real path of the vehicle effectively.

II. Sample Acquisition

The establishment of the SISM is based on the statistics of historical samples of the increment-state. The sample is consist of the vehicle state and the increment of the lateral error, as shown in Fig. 4, and collected in four steps as follows:

Step 1: Collect the planned trajectory points (green points) and the real path points (yellow points). The real path points are obtained from the Differential GPS/IMU integrated system with 10Hz frequency. The positioning error of the D-GPS is less than 0.1m.

Step 2: Find the specific point (the blue point) which is the intersection point of the real path and the normal line of the planned trajectory point (the green point). Then calculate the position of the specific point and the distance (the lateral error) between this specific point and its corresponding trajectory point.

Step 3: Calculate the velocities of the vehicle on specific points, according to the velocities on real path points and the distances between real path points and the specific points.

Step 4: Calculate the increment of the lateral error, which is the difference between two adjacent lateral errors.

Finally, we can obtain the increment-states, namely historical samples, from the planned trajectory and the real path of the vehicle in the historical dataset.

III. State-Increment Statistical Model

The State-Increment Statistical Model is built through analyzing historical samples of the increment-state. Suppose that the set of increment-states is defined as $X^* = (R, V, E, \Delta E)$, and the corresponding set of states is $X = (R, V, E)$. Then the increment-state is $x_k^* = (x_k, \Delta e_k) \in X^*$ and its corresponding state is $x_k = (r_k, v_k, e_k) \in X$, as shown in Fig. 4.

First, we analyze the value ranges of R, V, E, and discretize them via setting the intervals.

$$R = \bigcup_{i=1}^{N_R} R_i, \quad \text{and } \forall r \in R_i,\ r_i \leqslant r < r_{i+1}, \tag{1}$$

$$V = \bigcup_{j=1}^{N_V} V_j, \quad \text{and } \forall v \in V_j,\ v_j \leqslant v < v_{j+1}, \tag{2}$$

$$E = \bigcup_{l=1}^{N_E} E_l, \quad \text{and } \forall e \in E_l,\ e_l \leqslant e < e_{l+1}. \tag{3}$$

Then we cluster the increment-states via using K-means method which is introduced as the following steps.

Step 1: Suppose that $S_{i,j,l}$ represents the state subspace in which any state (r_k, v_k, e_k) satisfies $r_k \in R_i, v_k \in V_j, e_k \in E_l$. In every subspace $S_{i,j,l}$, one sample is randomly selected as the initial mean m_n where n is calculated based on the Eq. (4). If there is no state in $S_{i,j,l}$, m_n will be null. Then we will get the initial mean vector $(m_1, m_2, \ldots, m_N), N = N_R \times N_V \times N_E$.

$$n = (i-1) \times N_V \times N_E + (j-1) \times N_E + l. \tag{4}$$

Step 2: Suppose that $\hat{X}_n, n = 1, \ldots, N$ are the empty state sets and $I_n, n = 1, \ldots, N$ are the empty increment sets.

Step 3: Find the nearest mean, which is not null, for the state of any sample $x_k^* = (x_k, \Delta e_k)$ and assign the state x_k to the corresponding state set and the increment Δe_k to the corresponding increment set of the nearest mean, which are described in the Eqs. (5)-(7).

$$n^* = \arg\min_n \parallel x_k - m_n \parallel, \tag{5}$$

$$\hat{X}_{n^*} = \hat{X}_{n^*} \cup \{x_k\}, \tag{6}$$

$$I_{n^*} = I_{n^*} \cup \{\Delta e_k\}. \tag{7}$$

Step 4: Update the mean vector $(m_1, m_2, \ldots, m_N)$ which is calculated in the Eq. (8).

$$m_n = \frac{1}{\left|\hat{X}_n\right|} \sum_{x_k \in \hat{X}_n} x_k, \quad n = 1, \ldots, N. \tag{8}$$

Step 5: Repeat from Step2 to Step4 until the mean vector is changeless.

Finally, the state sets are clustered as described in Fig. 5. The final mean m_n can represent the mean of the samples in the subspace $S_{i,j,l}$. Corresponding to $S_{i,j,l}$, the gaussian curve is fitted for the increment set I_n, and estimate its mean μ_n and standard deviation σ_n. Consequently, the SISM is composed of the mean μ_n, the standard deviation σ_n, and the sample number N_n in each increment set I_n, corresponding to the state subspace $S_{i,j,l}$. If there is no state in a subspace,

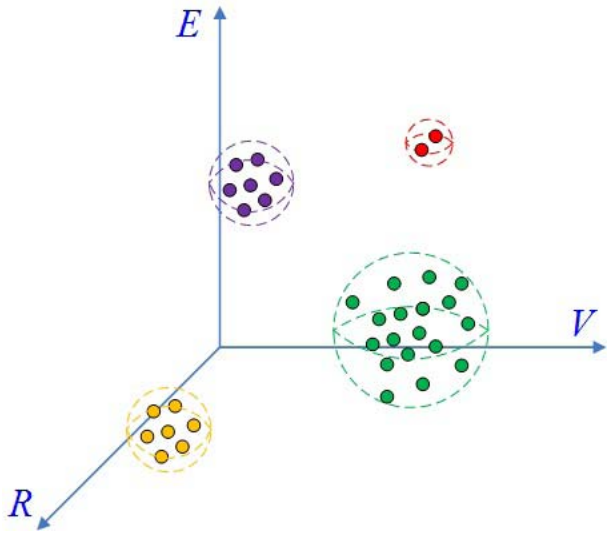


Fig. 5. The cluster sample sets are described by colorful balls. Every set consists of three intervals including the curvature radius, the velocity and the lateral error.

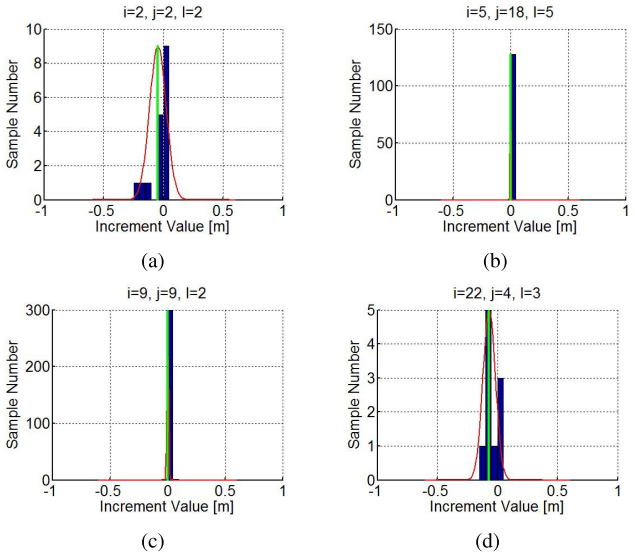


Fig. 6. These figures show the distributions of the sample number in different increment sets. The parameters, i, j, l, are the interval indexes of the curvature radius, the velocity and the lateral error. The red line is the fitting gaussian curve and the green line shows the mean of the lateral error increment in the set.

the corresponding increment set will be empty. As a result, the mean, the standard deviation and the sample number of lateral error increments will be null in this subspace.

In the experimental section of this paper, the SISM is built by 74904 historical increment-states which are collected before the autonomous vehicle competition "Future Challenge 2014". We divide the velocity range into 19 intervals from $1m/s$ to $10m/s$, the lateral error range into 6 intervals from $0m$ to $1.2m$ averagely, and the radius range into 31 intervals from $5m$ to $900m$ experientially. Then the increment-states are clustered to obtain the increment sets $I_n, n = 1, 2, \ldots, 3534 (= 31 \times 19 \times 6)$ for all state sets. Then the gaussian curve of the increments is fitted in each increment set, and some results are shown in Fig. 6. Finally, the SISM is built as shown in Fig. 7.

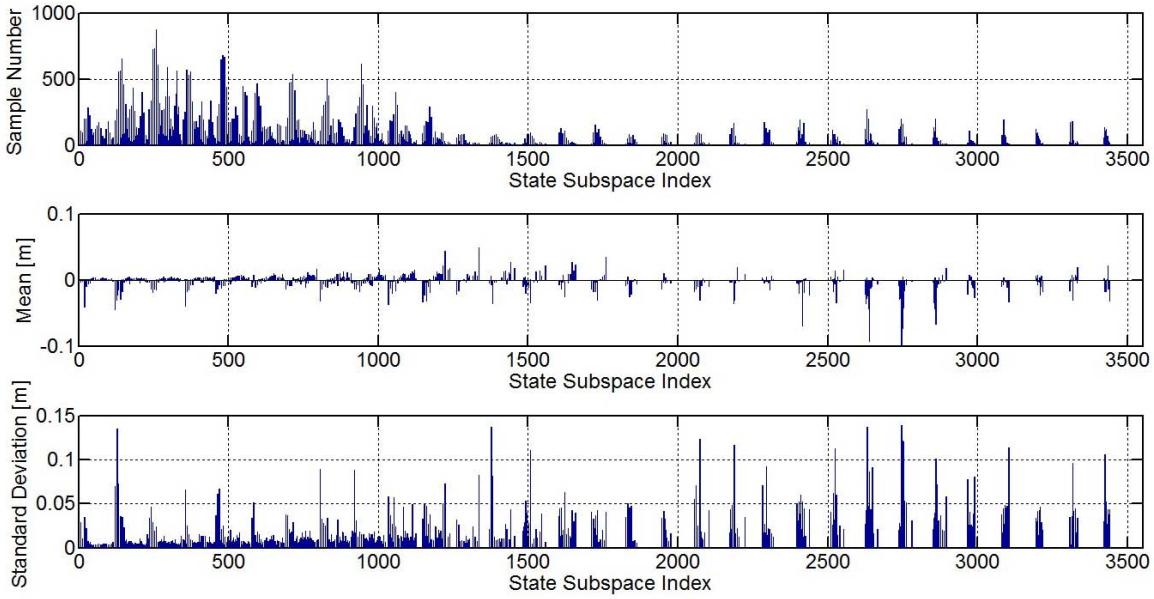


Fig. 7. The SISM is composed of three kinds of distribution of lateral error increments, including sample numbers, means and standard deviations, in all state subspaces.

A. Time Cost

Although it takes long time to build the SISM offline, the application of the SISM online is almost real-time. The time cost of the path prediction is less than 0.1s for a path about 50m long. When we obtain the current state to predict the lateral error increment, we will first calculate the interval indexes i, j, l of the curvature radius, the velocity and the lateral error based on the discretized interval setting. Then we can get the corresponding mean μ_n of increments in the state subspace $S_{i,j,l}$ via calculating n based on the Eq. (4). The path prediction is described detailedly in Section IV. In essence, the SISM works as a lookup table which can reduce the time cost of the calculation via increasing the memory cost.

Our computation platform includes a 2.4GHz processor with 6MB cache memory and 4.0GB RAM. The establishment of the SISM is performed on MATLAB R2010a, and its application is performed on Visual Studio 2010.

B. Performance Analysis

The accuracy of the SISM is depended on three factors. The first is the sample number in each state subspace. The more the sample number is in a state subspace, the more accurate the mean and the standard deviation of the lateral errors are in this subspace, as shown in Fig. 6. The second is the total number of subspaces. The third is the ratio of the invalid subspaces to the total subspaces. Apparently, there are many invalid means and standard deviations in corresponding subspaces in the SISM, as shown in Fig. 7, because there is no sample in these subspaces. These subspaces are called as the invalid subspaces. If the distribution of samples is limited and the subspace number is large, the number of invalid subspaces will be large. In order to improve the accuracy of the SISM, the samples should be as much as possible in each state subspace, the total number of subspaces should be as much as possible and the ratio of the invalid subspaces to the total subspaces should be as small as possible.

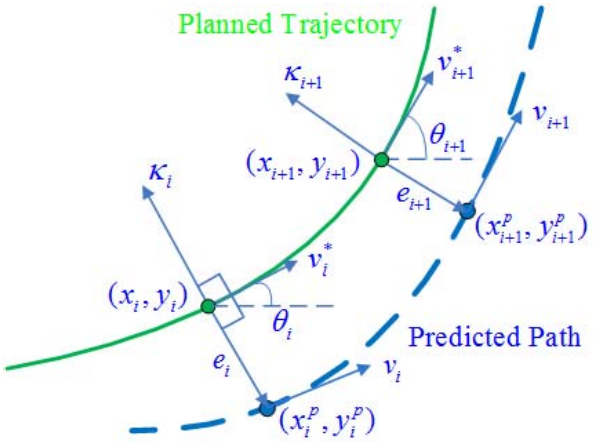


Fig. 8. The green points are the planned trajectory points and the blue points are the predicted points on the predicted path corresponding to the planned trajectory points.

IV. Path Prediction

In order to ensure the safety of the vehicle, it is necessary to predict its real path when the vehicle follows the planned trajectory. The SISM can be used to predict the real path accurately by analyzing the changing rule of the lateral error between the planned trajectory and the real path in history.

Suppose that the posture-velocity sequence of a planned trajectory is defined as (X, Y, Θ, K, V^*), and $\left(x_i, y_i, \theta_i, \kappa_i, v_i^*\right), i = 1, \ldots, M$ is the posture-velocity value on the i-th point of the trajectory, where $x_i \in X$ is the coordinate in X position, $y_i \in Y$ is the coordinate in Y position, $\theta_i \in \Theta$ is the tangent angle, $\kappa_i \in K$ is the curvature, $v_i^* \in V^*$ is the expected velocity, and M is the point number of this trajectory, as shown in Fig. 8. In the process of the path

prediction, the posture-velocity sequence $(x_i, y_i, \theta_i, \kappa_i, v_i^*)$ of the planned trajectory and the initial lateral error e_1 are the input, and the position sequence (x_i^p, y_i^p) of the predicted path is the output.

For the sake of avoiding the influence of the velocity tracking error, we assume that the vehicle velocity v_i is equal to the expected velocity v_i^*. Then the increment-state $(r_i, v_i, e_i, \Delta e_i)$ can be calculated from the Eqs. (9)-(11). Finally, the position (x_i^p, y_i^p) of the predicted path point can be obtained based on the Eqs. (12) and (13). The symbol variate, sym, is positive if the vehicle is on the left side of the planned trajectory and is negative if the vehicle is on the right side.

$$n = f(\frac{1}{\kappa_i}, v_i^*, e_i), \tag{9}$$

$$\Delta e_i = \begin{cases} 0, & \mu_n \text{ is null} \\ \mu_n, & \mu_n \text{ is not null,} \end{cases} \tag{10}$$

$$e_i = e_1 + \sum_{k=1}^{i-1} \Delta e_k. \tag{11}$$

$$x_i^p = x_i + e_i \cdot \cos\left(\theta_i + sym \cdot \frac{\pi}{2}\right), \tag{12}$$

$$y_i^p = y_i + e_i \cdot \sin\left(\theta_i + sym \cdot \frac{\pi}{2}\right). \tag{13}$$

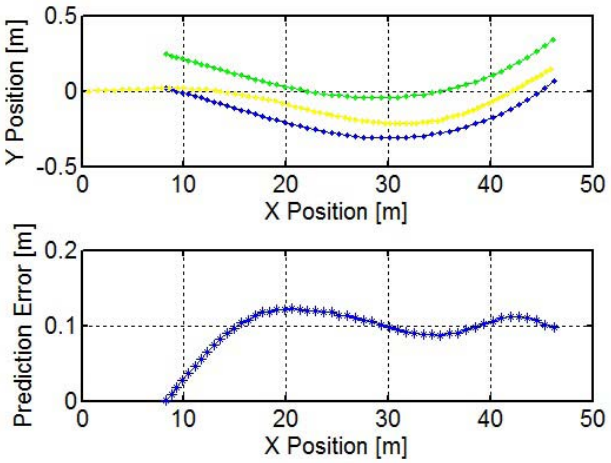
Fig. 9 presents the prediction result of the real path of the vehicle in the same traffic scene, which is shown in Fig. 3. The maximum prediction error between the real path and the predicted path is only $0.12m$. More experimental results are presented in Section V.

A. Comparison With Traditional Prediction

Traditional prediction methods predict the position sequence (the real path) of the vehicle based on the vehicle model, like bicycle model, without taking the controller and the actuator into account. These methods can predict the vehicle position according to the expected front wheel angle sequence and the expected acceleration sequence. However, from the whole of the autonomous vehicle, the motion constraints considered by traditional prediction methods are one-sided. Because the vehicle model is only a part of the factors that affect the trajectory following of a vehicle. Moreover, It is difficult to model the actuator, because the machine driven system usually has a time lag, as well as strong non-linearity.

Compared with the vehicle model, the SISM models the integration of the controller, the actuator and the vehicle model, as shown in Fig. 1. As a result, the prediction method based on the SISM can predict directly according to the planned trajectory and the vehicle state. It is difficult to compare the SISM-based method with traditional prediction methods, because the modeling objects in these two kinds of methods are different. The comparison of prediction results can not demonstrate which method is better. If we take the trajectory and the vehicle state directly as the inputs of traditional prediction methods, the traditional prediction results will be affected necessarily by the accuracy of the models of the controller and the actuator.

(a)



(b)

Fig. 9. The prediction result based on the SISM. The green line is the planned trajectory, the yellow line is the real path of the vehicle when it follows the trajectory, and the blue line is the predicted path. The prediction error (the blue line) from the predicted path to the real path is shown in (b).

V. Trajectory Evaluation

In traditional planning methods, some common criteria are utilized to search the best trajectory, such as feasibility, safety, smoothness, time cost, energy cost and so on. Among them, the feasibility of the trajectory is the most important one. All other criteria are meanless if the trajectory is unfeasible. Usually the traditional method to judge that the trajectory is feasible, is that the trajectory satisfies the kinematic constraints of the vehicle. In other words, the feasible trajectory can be followed by the vehicle with a tolerable lateral error. As a consequence, the optimal trajectory is the feasible one with the lowest cost in all trajectories.

However, human has an intuitionistic and high-efficient method to search the best trajectory. Instead of the optimal trajectory, human tends to regard the most familiar trajectory as the best one which has been successfully followed for many times in history. Obviously the familiar trajectory is feasible because the vehicle is able to follow it in history. For the sake of implementing this evaluation method, we establish the

experience cost functions (14)-(18) based on the SISM.

$$n = f(r_i, v_i, e_i), \tag{14}$$

$$c_i = \begin{cases} 1, & N_n = 0 \\ \dfrac{(M-i)\sigma_n}{N_n}, & N_n \neq 0, \end{cases} \quad i = 1, \ldots, M, \tag{15}$$

$$CC = \sum_{i=1}^{M} (c_i)^2, \tag{16}$$

$$\begin{aligned} TC &= \sum_{i=2}^{M} \left(exp(\frac{3i}{M}) \cdot e_i \right) \\ &= e_1 \sum_{i=2}^{M} exp(\frac{3i}{M}) + \sum_{i=1}^{M-1} \left(\mu_i \sum_{k=i+1}^{M} exp(\frac{3k}{M}) \right), \end{aligned} \tag{17}$$

$$EC = a_1 \cdot CC + a_2 \cdot TC, \tag{18}$$

where n is the point number of the planned trajectory, a_1 and a_2 are the weight coefficients, $exp(\cdot)$ is the natural exponent function. When the vehicle reaches the i-th point of this trajectory, we will get the i-th state (r_i, v_i, e_i) of the vehicle. μ_n and σ_n are the mean and the standard deviation of the distribution of lateral error increments in the state subspace where the i-th state is. Respectively, N_n is the sample number of increment-states of this state subspace.

A. The Credibility Cost - CC

The cost functions given in the Eqs. (15) and (16) describe the experience credibility of the predicted path. c_i is the sub-credibility-cost of the predicted position corresponding to the i-th trajectory state. The credibility cost CC is the credibility cost of the whole predicted path by accumulating the square of c_i.

Apparently, the accuracy of the predicted path is inversely proportional to the credibility CC. Because the predicted path is obtained by integrating the incremental mean of the lateral error, according to the statistic of the historical samples of the increment-state. The larger the number of the historical samples is and the smaller the standard deviation of the increment distribution is, then the more accurate the mean of the increment distribution is and the more credible the predicted path is. In addition, due to the influence of the increment integration, the cost weight of the former integrated increment is larger than the weight of the next one.

B. The Tracking Cost - TC

The cost function in the Eq. (17) describes the tracking accuracy of the vehicle according to the lateral error between the planned trajectory and the predicted path. The smaller the value of TC is, the more accurately the vehicle will follow the trajectory. $exp(\frac{3i}{n})$ is the cost weight of the lateral error on different trajectory points. Because the aim of following the trajectory is to make the vehicle reach the goal pose, the cost weight of the current error is larger than the weight of the former one. Additionally, the cost weight of the initial lateral error is zero as it is a known constant.

In the Eq. (11), Δe_i is the expected value of the lateral error increment on the i-th trajectory point, namely the mean of lateral error increments, μ_i. Therefore, according to the Eqs. (11) and (12), the lateral errors $e_i, i = 1, 2, \ldots, n$ on all trajectory points can be calculated by integrating $\mu_i, i = 1, 2, \ldots, n$. Then TC can be obtained based on the Eq. (17).

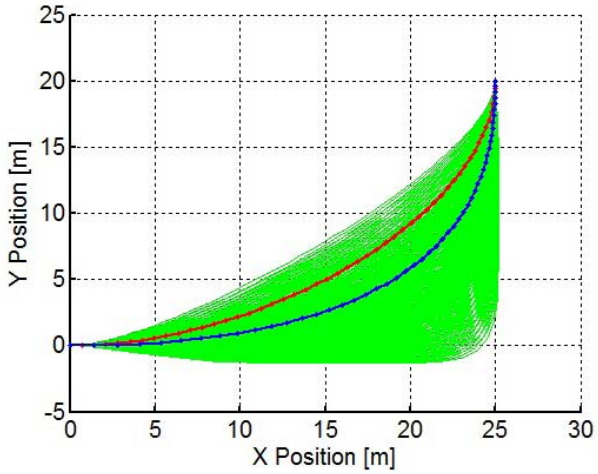


Fig. 10. The optimization result in the simulation. All the trajectories are generated based on the cubic Bezier curve. Among them, the red one is the best trajectory with the red velocity curve and the blue one is the best trajectory with the blue velocity curve in Fig. 10.

C. The Integrated Experience Cost - EC

The final experience cost C_{ex} is the integrated one between the credibility cost CC and the tracking cost TC. In the Eq. (18), a_1 and a_2 are the normalized coefficients because the magnitude orders of CC and TC are different. The equations to calculate a_1 and a_2 are as follows.

$$a_1 = \frac{1}{\sum_{j=1}^{m} CC_j}, \tag{19}$$

$$a_2 = \frac{1}{\sum_{j=1}^{m} TC_j}, \tag{20}$$

where m is the number of the planned trajectories.

Based on the Eq. (18), the final experience costs of all trajectories can be calculated and the best trajectory is the one with the lowest experience cost.

VI. Experiments and Discussions

A. Experiment in the Simulation

In the simulation, the initial pose of vehicle is $(0, 0, 0)$ and the goal pose is $\left(25, 20, \frac{\pi}{2}\right)$. Then many cubic Bezier trajectories can be generated from the initial pose to the goal pose, as shown in Fig. 10. For the sake of comparing different best trajectories with different velocity curves, we set two expected velocity curves when the vehicle follows the trajectories. The red one is a uniform velocity curve and the blue one is an accelerative velocity curve, which are described in Fig. 11.

Fig. 12 and Fig. 13 show two distributions of the credibility costs and the tracking costs for all the trajectories which are calculated by the Eqs. (15)-(17). Then according to two expected velocity curves, we can search two different best

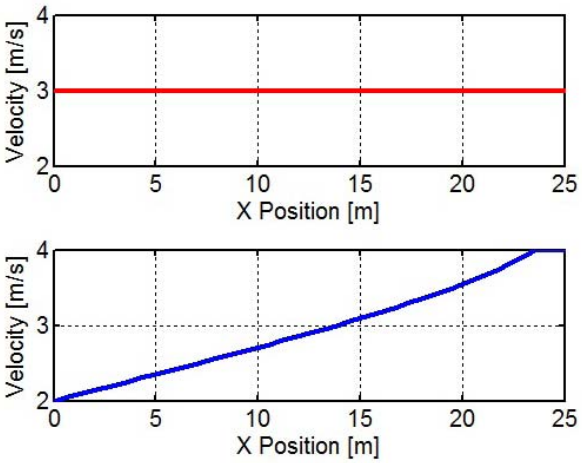


Fig. 11. There are two expected velocity curves in the simulation. The red curve is a uniform velocity sequence and the blue one is an accelerative velocity sequence. In order to be consistent with Fig. 9, we take the X position as the abscissa.

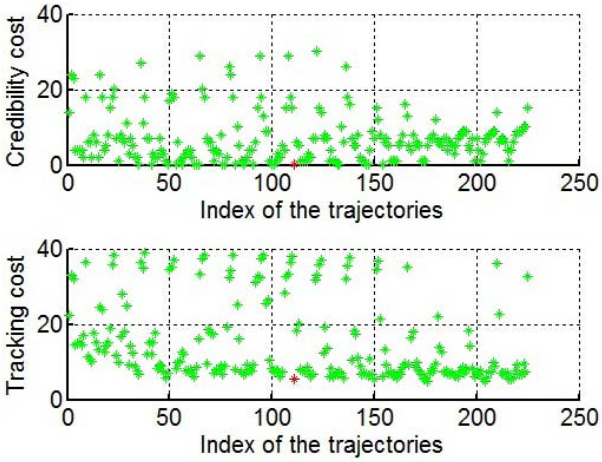


Fig. 12. The distributions of the credibility cost and the tracking cost for all the trajectories with red velocity curve. (The red stars show the costs of the best trajectory.)

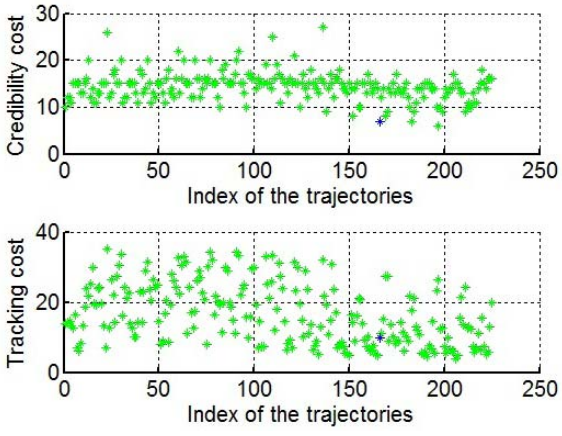
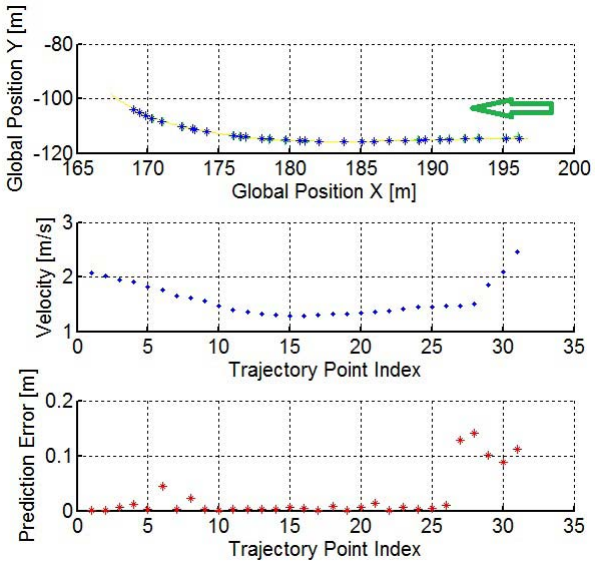


Fig. 13. The distributions of the credibility cost and the tracking cost for all the trajectories with blue velocity curve. (The blue stars show the costs of the optimal trajectory.)

trajectories (the red one and the blue one) with the lowest experience cost based on the Eqs. (18)-(20).

(a)



(b)

Fig. 14. The planning and analysis results when the vehicle turned right at an intersection.

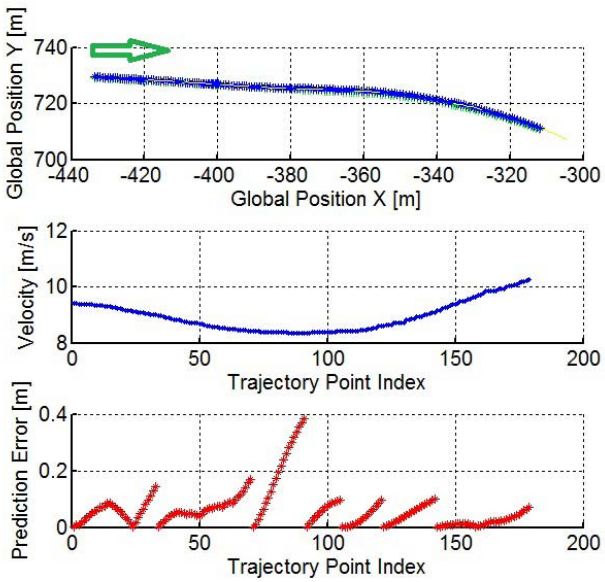
In addition, every trajectory is composed of the same number of points, we can obtain the best expected velocity curve through comparing the experience costs of the red trajectory and the blue trajectory. It is obvious that the experience cost of the red trajectory is lower than the experience cost of the blue one. In other words, the vehicle used to turn left at a uniform speed rather than at a variable speed.

B. Experiments in Real Scenarios

In the autonomous vehicle competition "Future Challenge 2014" which was held in Changshu, China, the competition data which are collected from real urban scenarios, are used to do a offline test to verify the effectiveness of the SISM. These testing scenarios include crossing the bridge, keeping the curve lane, passing the roundabout and right turning at the intersection and so on. Some prediction results are shown as follows.

In order to compare the prediction result with the real scenario and the planned trajectory, the predicted path and the real

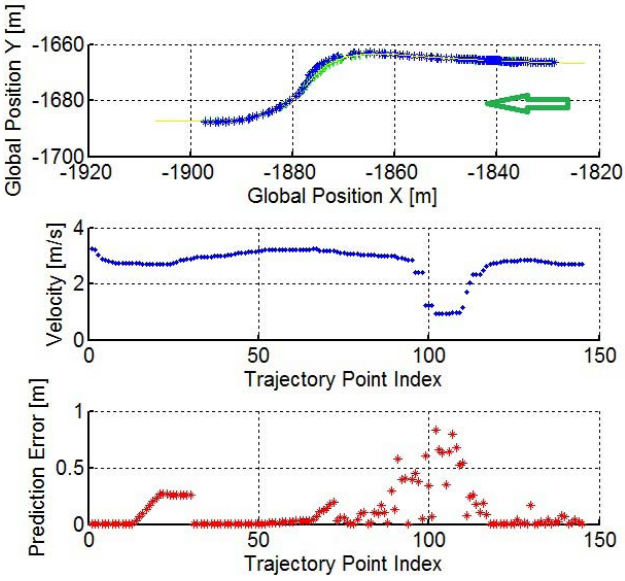
(a)



(b)

Fig. 15. The planning and analysis results when the vehicle crossed a bridge.

(a)



(b)

Fig. 16. The planning and analysis results when the vehicle passed a roundabout.

path are mapped from the Cartesian space to the corresponding image space by using IPM reverse transformation. As shown in the following scenario images (Figs. 14-16 (a)), the green line is the planned trajectory, the blue line is the predicted path and the yellow line is the real path of the vehicle. Each following analysis figure (Figs. 14-16 (b)) have three subfigures. The top one shows the planned trajectory, the predicted path, the real path and the moving direction (the green arrow) in the global coordinate. The medial one gives the real velocity sequence. The bottom one gives the prediction error sequence between the predicted path points and the corresponding real path points when the vehicle followed the planned trajectory. Three typical urban scenarios in the competition are listed as follows.

1) Turning Right at an Intersection: The predicted path, the real path and the planned right-turning trajectory at a intersection are plotted in Fig. 14(a). The results are displayed in Fig. 14(b), including the planning and prediction results in the Cartesian coordinate, the velocity curve and the prediction error between the predicted path and the real path. These results show that the prediction error is nearly zero when the vehicle moved along the high-curvature trajectory at the speed under $2m/s$.

2) Crossing the Bridge: Fig. 15 shows the planning and prediction results when the vehicle crossed a bridge. In this process, the vehicle moved at a speed above $8m/s$, as shown in Fig. 16(b). From the third figure in Fig. 16(b), it is apparent that the prediction error line is composed of several upward curves which are caused by the replanning. Because the bridge is an arch structure, the statistical model is less accurate than the one on a level road. As a consequence, the prediction error is rising slowly in a planning frame and the maximum error appeared when the vehicle moved nearly to the top of the bridge, as shown in the third image in Fig. 16(a).

3) Passing the Roundabout: Fig. 16 shows the process that the vehicle passed a small roundabout in a school. From the analysis results, there is a large prediction error in this scenario. In the urban scenarios, the similar high-curvature

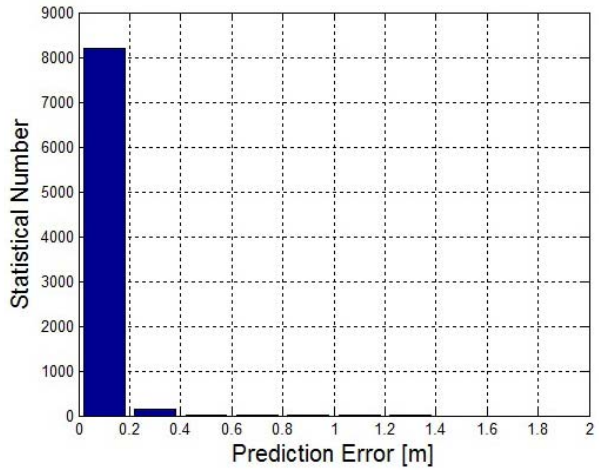


Fig. 17. The statistical result of the prediction errors in the competition.

trajectory is always generated in small intersections, as shown in Fig. 14, and the velocity is always below $2m/s$ when the vehicle follows this kind of trajectories. Therefore, most high-curvature samples are collected when the velocity is below $2m/s$. As a result, the real path is nearly the same as the planned trajectory, but the prediction error between the predicted path and the real path is too large because of the inadequate number of samples. However, the prediction error in this scenario will decrease rapidly with the increase of the number of same samples.

In addition, the prediction error is nearly zero at the end of the process because of two reasons. One is that the replanning can eliminate the large prediction error which is generated by the increment integration. The other one is that the curvature in the last part of the trajectory is similar to the one in the initial part and that the SISM is accurate enough in this state subspace.

4) Final Statistical Result: Throughout the competition, the number of valid trajectory points, which are followed by our vehicle, is 8374. Fig. 17 shows the statistical result of the prediction errors which are the distances between the predicted path points and the corresponding real path points of our vehicle. Apparently the rate of the prediction error, which is less than $0.2m$, is as high as 97.7%. Moreover, this rate will increase when more samples of the increment-state are added to improve the accuracy of the SISM.

VII. Conclusion and Future Work

A. Conclusion

Compared with traditional kinematic model of the vehicle, the state-increment statistical model(SISM) can be utilized to evaluate the trajectory more efficiently, predict the real path of the vehicle more accurately, and navigate the vehicle more human-likely. There are three reasons as follows.

1) Self-Improvement: The SISM is built by the statistics of all the historical samples of the increment-state. It can be improved by collecting more samples in different subspaces, increasing the number of subspaces and reducing the ratio of the invalid subspaces to the total subspaces.

2) Adaptation: Compared with the vehicle model, the SISM can describes the motion performance of a whole vehicle, including the controller, the actuator and the kinematic model. No matter what kind of controller and actuator, the SISM can be built and used to predict the vehicle state accurately in real time. In addition, the SISM can also be used to design the cost function for different trajectory generation methods.

3) Experience Based Planning: The traditional planning algorithm is always an environment-based planning which searches the most optimal solution in current environment. Generally, the criteria for the optimal solution include smoothness, shortest time, fastest speed and so on. But human will not consider these criteria when they make driving decision. They tend to drive with a familiar way in similar traffic scenes. As same as human decision, the SISM-based planning, which is a kind of experience-based planning, searches the most familiar solution in current environment. The experience cost functions will be used to search the most familiar trajectory as the best one which is mentioned in Section IV. Therefore, the experience-based planning is like human driving planning to make a behavior habitually.

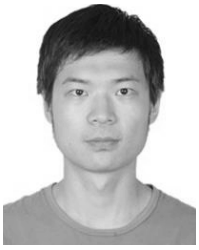
B. Future Work

In the SISM, the velocity errors between the expected velocity sequence and the real velocity sequence, are ingored. In future work, we will improve the SISM via taking the velocity error increment and the lateral error increment as outputs. Based on the improved SISM, the real velocity and the real position can be predicted simultaneously. Then the real trajectory can be obtained. As a consequence, we can predict the vehicle motion at anytime and optimize the planned trajectory inversely.

References

[1] J. Leonard *et al.*, "A perception-driven autonomous urban vehicle," *J. Field Robot.*, vol. 25, no. 10, pp. 727–774, 2008.
[2] I. Miller *et al.*, "Team Cornell's Skynet: Robust perception and planning in an urban environment," *J. Field Robot.*, vol. 25, no. 8, pp. 493–527, 2008.
[3] M. Montemerlo *et al.*, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
[4] A. Bacha *et al.*, "Odin: Team VictorTango's entry in the DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 467–492, 2008.
[5] J. Bohren *et al.*, "Little ben: The ben franklin racing team's entry in the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 598–614, 2008.
[6] S. Kammel *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 615–639, 2008.
[7] B. Nagy and A. Kelly, "Trajectory generation for car-like robots using cubic curvature polynomials," *Field Service Robotics (FSR'01)*, Jun. 2001.
[8] A. Kelly and B. Nagy, "Reactive nonholonomic trajectory generation via parametric optimal control," *Int. J. Robot. Res.*, vol. 22, nos. 7–8, pp. 583–601, Jul. 2003.
[9] C. Urmson *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.
[10] D. Ferguson, T. M. Howard, and M. Likhachev, "Motion planning in urban environments," *J. Field Robot.*, vol. 25, nos. 11–12, pp. 939–960, Nov. 2008.
[11] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét frame," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2010, pp. 987–993.

[12] M. Werling, S. Kammel, J. Ziegler, and L. Gröll, "Optimal trajectories for time-critical street scenarios using discretized terminal manifolds," *Int. J. Robot. Res.*, vol. 31, no. 3, pp. 346–359, 2011.
[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[14] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2013, pp. 5041–5047.
[15] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961–1976, Aug. 2015.
[16] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
[17] P. Qu, J. Xue, L. Ma, and C. Ma, "A constrained VFH algorithm for motion planning of autonomous vehicles," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jul. 2015, pp. 700–705.
[18] T. M. Howard, R. A. Knepper, and A. Kelly, "Constrained optimization path following of wheeled robots in natural terrain," in *Proc. Int. Symp. Experim. Robot.*, 2008, pp. 343–352.
[19] W. Yao, H. Zhao, F. Davoine, and H. Zha, "Learning lane change trajectories from on-road driving data," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2012, pp. 885–890.
[20] T. Gu and J. M. Dolan, "Toward human-like motion planning in urban environments," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun. 2014, pp. 350–355.
[21] B. Kim and K. Yi, "Probabilistic and holistic prediction of vehicle states using sensor fusion for application to integrated vehicle safety systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2178–2190, May 2014.
[22] J. Hardy, F. Havlak, and M. Campbell, "Multi-step prediction of nonlinear gaussian process dynamics models with adaptive Gaussian mixtures," *Int. J. Robot. Res.*, vol. 3, no. 9, pp. 1211–1227, 2015.
[23] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1764–1771.
[24] M. Greytak and F. Hover, "Planning to learn: Integrating model learning into a trajectory planner for mobile robots," in *Proc. Int. Conf. Inf. Autom. (ICIA)*, Jun. 2009, pp. 18–23.
[25] C. Ma, J. Yang, J. Xue, Y. Liu, and L. Ma, "State-statistical model based trajectory-band planning in urban environment," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jul. 2015, pp. 400–405.



Chao Ma received the B.S. degree in automatic control engineering from Xi'an Jiaotong University, Xi'an, China, in 2010.

He is currently pursuing the Ph.D. degree with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China. His research interests include statistical learning on the motion planning and the control for autonomous driving.

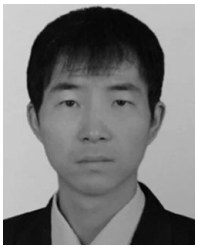
Jianru Xue (M'06) received the B.S. degree from Xi'an University of Technology in 1994, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2003, respectively. He was with Fuji Xerox, Tokyo, Japan, from 2002 to 2003, and visited University of California, Los Angeles, from 2008 to 2009.

He is currently a Professor with Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research field includes computer vision, visual navigation, and video coding based on analysis.

He served as a PC member of pattern recognition 2012, Asian Conference on Computer Vision in 2010 and 2012, respectively. He also served as a Co-Organization Chair of Asian Conference on Computer Vision 2009 and Virtual System and Multimedia 2006.

Yuehu Liu (M'06) received the B.E. and M.E. degrees from Xi'an Jiaotong University, China, in 1984 and 1989, respectively, and the Ph.D. degree in electrical engineering from Keio University, Japan, in 2000. He is currently a Professor with Xi'an Jiaotong University. His research is focused on computer vision, AR, and HCI. He is a member of the IEICE.

Jing Yang received the B.Eng. and M.Eng. in control science and engineering and the Ph.D. degree in pattern recognition and intelligent systems from Xian Jiaotong University, China, in 1999, 2004, and 2010, respectively. Since 1999, she has been a Lecturer with the Automation Department, Xian Jiaotong University. She is currently a member of project team on intelligent vehicles with the Institute of Artificial Intelligence and Robotics, Xian Jiaotong University. Her main research interests include autonomous vehicle control, intelligent control, robot motion planning, and neural networks.



Yongqiang Li received the B.S. degree in automation from Yanshan University, Qinhuangdao, China, in 2008. He started his postgraduate study in 2011 and received the M.S. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2014. He is currently pursuing the Ph.D. degree with the Visual Cognitive Computing and Intelligent Vehicle Laboratory, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His current research interests include motion planning and decision making in intelligent vehicles and autonomous driving.

Nanning Zheng (SM'94–F'06) received the degree from the Department of Electrical Engineering, Xi'an Jiaotong University, Xi'an, China, in 1975; the M.E. degree in information and control engineering from Xi'an Jiaotong University, Xi'an, in 1981; and the Ph.D. degree in electrical engineering from Keio University, Japan, in 1985.

He is currently a Professor and the Director with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, computational intelligence, image processing, and hardware implementation of intelligent systems.

He became a member of the Chinese Academy Engineering in 1999. Since 2000, he has been the Chinese representative on the Governing Board of the International Association for Pattern Recognition. He presently serves as an Executive Editor of the Chinese Science Bulletin.