

Research Article

Stochastic Parameter Identification Method for Driving Trajectory Simulation Processes Based on Mobile Edge Computing and Self-Organizing Feature Mapping

Jingfeng Yang,^{1,2,3} Zhiyong Luo ,³ Nanfeng Zhang,⁴ Jinchao Xiao,^{1,2} Honggang Wang,⁵ Shengpei Zhou,^{1,2} Xiaosong Liu,⁶ and Ming Li ^{7,8}

¹Shenyang Institute of Automation Chinese (Guangzhou) Academy of Sciences, Guangzhou 511458, China

²Shenyang Institute of Automation Chinese Academy of Sciences, Shenyang 110016, China

³School of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou 510006, China

⁴Huangpu Customs District Technical Center, Guangzhou 510730, China

⁵School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China

⁶Guangdong Zhongke Zhenheng Information Technology Co., Ltd, Foshan 528225, China

⁷South China Agricultural University, Guangzhou 510642, China

⁸Yaz Technology Co., Ltd., Guangzhou 510630, China

Correspondence should be addressed to Zhiyong Luo; 13925121031@139.com and Ming Li; liming@yazcn.com

Received 16 August 2020; Revised 4 November 2020; Accepted 14 December 2020; Published 7 January 2021

Academic Editor: Rongxin Cui

Copyright © 2021 Jingfeng Yang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid development of sensor technology for automated driving applications, the fusion, analysis, and application of multimodal data have become the main focus of different scenarios, especially in the development of mobile edge computing technology that provides more efficient algorithms for realizing the various application scenarios. In the present paper, the vehicle status and operation data were acquired by vehicle-borne and roadside units of electronic registration identification of motor vehicles. In addition, a motion model and an identification system for the single-vehicle lane-change process were established by mobile edge computing and self-organizing feature mapping. Two scenarios were modeled and tested: lane change with no vehicles in the target lane and lane change with vehicles in the target lane. It was found that the proposed method successfully identified the stochastic parameters in the process of driving trajectory simulation, and the standard deviation between simulation and the measured results obeyed a normal distribution. The proposed methods can provide significant practical information for improving the data processing efficiency in automated driving applications, for solving single-vehicle lane-change applications, and for promoting the formation of a closed loop from sensing to service.

1. Introduction

With the development of automated driving technologies, especially the onset of 5G technology, the demand for traffic data collection in the field of intelligent networking has increased significantly. Conventional driving environment detection technologies use road monitoring equipment with different traffic sensing technologies including video, GPS, geomagnetism, and radar. These devices are installed on roads by local public security agencies to ensure a safe and

smooth traffic flow, thus strengthening the law enforcement against traffic violations and the intelligent control of city-level transportation. However, these driving environment detection technologies cannot satisfy automated driving communication requirements of short time delay, high reliability, wide coverage, and vehicle-to-vehicle communication. Under this backdrop, the application of electronic registration identification of motor vehicles has been proposed and promoted on a national level through such measures as establishing national standards. Electronic

registration identification of motor vehicles have been promoted in a few cities and applied to several areas such as smart parking [1, 2], intelligent signal control [3, 4], intelligent connected vehicles [5], special vehicle traffic management and special vehicle identification monitoring [6], and traffic operation supervision and environmental protection restrictions [7]. However, as a means of law enforcement, the application of vehicle interactions is still lacking due to different technical problems; thus, it is difficult to achieve the required accuracy for vehicle driving trajectory identification. Hence, researchers have proposed different identification methods to solve this problem; for example, grid-based methods have been successfully used in multiobjective optimization algorithms [8–10], such as the positioning of driverless cars [11–13]. Grewe et al. [14] explored a series of MEC-enabled, high-quality, and reliable vehicle-borne services (such as electronic horizon, which assist vehicle movement), summarized the challenging problems encountered in the application of MEC technology to the network of vehicles, and proposed some potential solutions. Truong et al. [15] advocated the use of a combination of software-defined networking and MEC technology to address the problems of existing vehicle-borne self-organizing networks, such as inadequate coverage of communication range, unscalable network communication capacity, and a management that lacks intelligence and flexibility. They designed an edge computing network framework (dubbed FSDN) for the Internet of vehicles and MEC server that acts as a distributed zone controller to provide different local services, such as streaming of media content distribution and vehicle lane-change prompts, aimed at providing excellent decision-making capabilities. Real-time and accurate traffic information is the prerequisite for an improved urban traffic efficiency; Dikaiakos et al. [16] proposed the vehicle information transmission protocol (VITP) and implemented dynamic route planning using the forward traffic status and roadside points of interest obtained from query requests to downstream vehicles through the geographic routing protocol. Dombush and Joshi [17] studied the automatic discovery mechanism of traffic congestion and the distributed clustering of abnormal traffic flow and achieved the aggregation of perceived data through interactive sensing between vehicles. Zhang and Zhao [18] constructed a mathematical model to automatically collect, aggregate, process, and transmit traffic information and dynamically update the traffic coverage of the entire road network. Gramaglia et al. [19] used external data sources such as vehicle beacon messages and weather conditions to detect the degree of traffic congestion through complex event processing. Terroso-Saenz et al. [20] and Li et al. [21] detected traffic congestion from intervehicle collaboration where each vehicle used the collected beacon messages to estimate the surrounding traffic congestion using fuzzy logic and then corrected the individual estimation error through consultation. Bauza et al. [22] conducted traffic condition sensing through VANET collaboration and made short-term traffic condition forecasts using a linear least squares method. However, these studies have only solved the problems of cooperative traffic condition sensing and

neither consider implementation issues in automated driving scenario nor analyze important driving trajectories in vehicle-to-vehicle communication. In the present paper, based on electronic registration identification of motor vehicles and readers, a closed loop from perception to service was formed using mobile edge computing and self-organizing feature mapping to identify stochastic parameters for driving trajectory simulation processes and solve different application problem of lane-change scenarios in automated driving.

2. Driving Behavior, Speed, and Trajectory Calculation Based on Electronic Registration Identification of Motor Vehicles

The vehicle data collection method integrated with electronic registration identification of motor vehicles reads the vehicle information of the vehicle from the on-board unit with the electronic registration identification of motor vehicles reader installed on the gantry frame. The collected information is then uploaded to the comprehensive sensing base station on the test section of the road for edge computing. The overall deployment of the comprehensive sensing base station can complete data collection, verification, transmission, and processing in actual applications. The overall layout and actual installation are shown in Figure 1:

The main method of acquiring the driving speed of a vehicle with an electronic registration identification of motor vehicles reader uses the UHF radio frequency identification technology. The directional horizontally-polarized UHF antenna of the registration reader interacts with the unit installed on the front windshield of the vehicle and calculates the speed of the vehicle by measuring the time difference for the vehicle to pass through a fixed distance in the identification zone. Specifically, the positions of the vehicle at different times are calculated from RSSI values returned by the electronic registration identification of motor vehicles reader at different times, and the travel speed can then be calculated from the time differences. By analyzing RSSI values returned by the vehicle after entering the identification cross-section, it can be determined whether the location of the vehicle is at the boundary between the direct illumination zone and the blind zone or at the farthest point in the reflection zone; subsequently, the vehicle speed is calculated by the formula $V = S/(T_o - T_i)$.

3. Mobile Edge Computing and Self-Organizing Feature Mapping

3.1. Mobile Edge Computing. The fundamental concept of mobile edge computing is to move the cloud computing platform to the edge of the mobile access network in an attempt to deeply integrate a cellular network of conventional telecommunication with Internet services and also to reduce the end-to-end delay of mobile service delivery. It changes the state of a separate network in conventional wireless communication systems by exploiting the inherent capabilities of wireless networks and improves the user

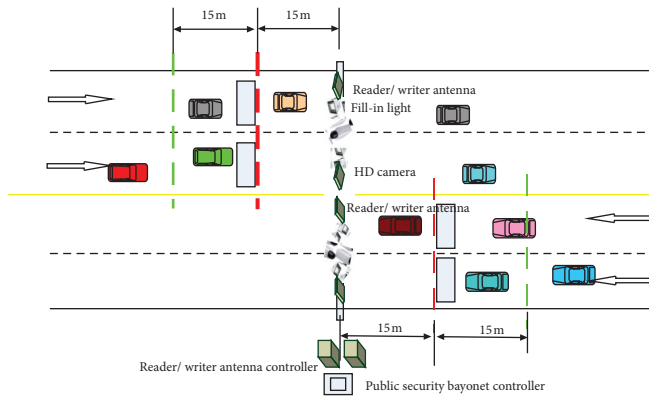


FIGURE 1: Comprehensive layout diagram and actual installation figure.



experience. By adding mobile edge computing platform network elements to a conventional wireless network, a business platform including contents, services, and applications, can be lowered to the edge of a mobile network to provide computing and data storage services to users. The basic characteristics of mobile edge computing include service localization, short-distance, low time-delay service delivery, user location awareness, and other network service capabilities. It can, therefore, bring new changes to the mode of operation for telecom operators and establish a new ecosystem of industrial chains and networks.

The edge computing architecture relies on the communication infrastructure and services provided by Edge Cloud Collaboration and LET/5G. The edge side mainly includes vehicle-borne edge computing units and RSU or MEC servers. Vehicle units depend on the cooperation of RSU or MEC servers. On the vehicle side, the control systems of intelligent networked vehicles are becoming increasingly complex with the use of AI and V2X applications. Load integration and simplified control systems are used to integrate different systems into HMI, and complex vehicle-borne information services including ADAD, IVI, digital instruments, and HUD are made to run on the same hardware platform through simulation technology. Moreover, load integration based on virtualization and the hardware abstraction layer HAL are more amenable to flexible business orchestration, deep learning model update, and software and firmware upgrade for a cloud-based vehicle driving system.

Due to several hardware constraints, the generation of a large amount of V2X data, and the imbalance of task processing, real-time scheduling is necessary for computing tasks. In the present work, a hybrid critical computing task scheduling technique was employed. The proposed method is linearly complex but is low cost and highly schedulable and can lower the time consumption of a system by reducing the number of repeated calculations in the passive delay priority updating. By establishing a more accurate upper bound of the peak period based on the quantitative relationship between the task number and the peak period, the spatial overhead of the system can be reduced and the scheduling performance can be improved.

3.2. Stochastic Parameter Identification Method Based on Self-Organizing Feature Mapping. According to the current practice of data optimization through neural networks, a self-organizing feature map (SOFM) was used in the current work to perform data optimization. Self-organizing feature map (SOFM) is a neural network with a clustering function [23]. Self-organizing mapping is a multidimensional scaling method, which can map any dimensional data in the input space to an output space of lower dimension. A self-organizing map (SOM) is an array of neurons at a regular location. Neurons can be placed on grid nodes in one, two, or higher dimensions, however usually in a one-dimensional or two-dimensional grid.

The operation of a SOFM network can be divided into two stages: training and work [24, 25]. In the training stage, samples from the training set are randomly fed into the network. For a specific input mode, the output layer generally has a neuron that produces the largest response to win. At the beginning of the training phase, it is uncertain which neuron in the output layer will respond most to the input mode. When the input mode changes to a different category, the winning neuron in the two-dimensional plane also changes. Neurons surrounding the winner neuron have a greater response due to the lateral mutual excitation. Therefore, weight vectors connected to the winner neuron and its surrounding neurons adjust in different degrees to the direction of the input vector. The degree of adjustment gradually diminishes as the distance from the winner neuron to its surrounding neurons increases. The network adjusts weights in a self-organizing manner by a large number of training samples. Finally, each neuron in the output layer becomes a sensitive neuron for a specific mode, and the corresponding inner weight vector becomes the center vector of each input mode. When the features of two input modes approach each other, neurons representing these two mode categories also approach each other, resulting in the formation of an ordered feature map reflecting the distribution of sample modes in the output layer.

After the completion of SOFM training, the specific relationship between each neuron in the output layer and each input mode category is completely determined, to use

the network as a mode classifier. At each input mode, the specific neuron in the output layer of the network generates the maximum response and automatically classifies the input. It should be noted that when the input mode of the network does not belong to any of the modes of the training set, SOFM can only assign it as belonging to the closest mode category. The classification process is performed by the following steps [24, 25].

In the first step, the winner neuron is found, the dot product of the input mode and the weight vector are calculated, and the winner neuron with the largest dot product is selected.

In the second step, the winning neighborhood is defined and the weight adjustment domain at time t centered around the winner neuron is determined. Generally, the initial neighborhood N is larger and gradually shrinks with the increase of the training time during the training process.

In the third step, the weights of all neurons in the winning neighborhood are adjusted by the following formula:

$$w_{i,j}(t+1) = w_{i,j}(t) + \eta(t, N) [x_i^p - w_{i,j}(t)], \quad (1)$$

$$i = 1, 2, \dots, n \quad j \in N_j \cdot (t),$$

where $\eta(t, N)$ is a function of the training time t and the topographical distance N between neuron j and the winner neuron j^* in the neighborhood.

Many functions can meet the above rules, for example, the following function can be constructed:

$$\eta(t, N) = \eta(t) e^{-N}. \quad (2)$$

The fourth step is the termination of the inspection. Unlike a BP network, the concept of output errors does not exist during the training of a SOFM network. As it is an unsupervised training, the training ends when the rate of learning $\eta(t)$ decays to zero or a certain preset positive small number. If the condition is not met, the process returns to step 1.

Kalman filter, based on the model of a state space, estimates the recursive relationship of an algorithm for a given process of filtering the state vector and constructs a measurement function (structural parameter or state) as an independent variable for an unknown quantity. It therefore derives an estimate of the unknown quantity from the measurement data. Kalman filter is widely applied to system identification. The conventional linear Kalman filter generally suffers from several shortcomings, such as low accuracy, poor stability, and slow response to target maneuver. In the application, the nonlinear motion equation of motion of the estimated dynamic system can be linearized and applied to the convergence of Kalman filter estimation. Furthermore, considering the nonlinear characteristics of the dynamic system near the reference trajectory and the comparison between strategically estimated algorithms, the real system can be described by a linearized equation.

Kalman filter can be divided into two parts: state prediction equation (state prediction) and state correction equation (observation update). The state prediction equation

is responsible to predict current state variables and estimate the error covariance in a timely manner to construct a priori estimate for the next time state. The state correction equation is responsible for giving feedback and combines the a priori estimate and the new observed variable to construct an improved a posteriori estimate.

Let the following equations describe an n -dimensional linear dynamic system and an m -dimensional linear observation system [26–28]:

$$\begin{aligned} x_{i+1} &= Ax_i + w_i, \\ z_i &= Hx_{i+1} + v_i, \end{aligned} \quad (3)$$

where x_{i+1} is an n -dimensional vector that represents the state of the system at the time instance $i+1$ and A is a $n \times n$ ordered matrix, which becomes the state transition matrix of the system, and reflects the state transition of the system from the i th sampling time to the $i+1$ st sampling time. w_i is an n -dimensional vector and represents the random interference acting on the system at the time instance $i+1$, and it is assumed that w_i is a Gaussian white noise sequence with a known zero mean and a covariance matrix Q_i , z_i is an m -dimensional observation vector, and H is a $m \times n$ ordered observation matrix and represents the transition from the state x_{i+1} to the observation vector z_i . For the m -dimensional observation noise v_i , it is also assumed that v_i is a Gaussian white noise sequence with a known zero mean and a covariance matrix R_i .

Therefore, the following recursive formula of the filter can be obtained:

$$\begin{aligned} K_i &= \hat{P}_i H^T (H \hat{P}_i H^T + R_i)^{-1}, \\ \hat{P}_i &= A P_{i-1} A^T + Q_i, \\ P_i &= (I - K_i H) \hat{P}_i, \\ \hat{x}_{i+1} &= x_{i+1} + K_i (z_i - H x_{i+1}), \end{aligned} \quad (4)$$

where Q_i is an $n \times n$ -ordered covariance matrix of the model noise w_i , R_i is the $m \times m$ -ordered covariance matrix of the observed noise v_i , K_i is the $n \times m$ -ordered gain matrix, \hat{x}_{i+1} is an n -dimensional vector and represents the estimated value after filtering at the time instance $i+1$, and P_i is the estimated error covariance matrix of order $n \times n$.

According to the calculation method described above, one can start with x_1 and a given P_0 and then recursively calculate the estimate of each time state by the known matrices Q_i , R_i , H , and A and the observation value z_i at the time i . If the linear system is stationary, then A and H are both constant matrices. If the model noise w_i and the observation noise v_i are stationary random sequences, then Q_i and R_i are constant matrices. Under such conditions, the constant gain discrete Kalman filter becomes asymptotically stable.

4. Tests and Results

In the present work, a single-vehicle changing lane was used as the main application scenario to establish a motion model and perform system identification. The single-

vehicle lane-change process was further divided into two scenarios: (i) lane change with no vehicle in the target lane and (ii) lane change with vehicles in the target lane. The trajectory of the lane-change execution phase began as the vehicle started to move from the initial lane to the target lane, and it proceeds to cross the lane line and reaches the target lane. In the entire process of crossing the lane line, any turning movement of the vehicle was considered as a failure of the execution of the lane-change operation. As the lane line information and the motion trajectory in the environment were acquired by vehicle-borne units and roadside units and calculated by edge computing, the distance between the vehicle and the lane line was calculated directly at any time.

The vehicle trajectory was expressed by the mathematical expression $\eta(t) = (X(t), Y(t), dX/dt(t), dY/dt(t))$ to include the horizontal and vertical positions and speed information of the vehicle in the observation geodetic coordinate system, where X, Y , and t represent, respectively, the horizontal coordinate, vertical coordinate, and sampling time. The schematic diagram of a lane-change process is shown in Figure 2.

Let t_0 be the running speed of the vehicle at time v_0 ; then, its lateral displacement and longitudinal position are, respectively, $L_h = v_0 t_0 \cos\theta$ and $L_z = v_0 t_0 \sin\theta$. Starting from the initial moment of the trajectory, a search could find two points with the smallest time intervals on both sides of the lane line. Now, setting them to t_{c1} and t_{c2} , respectively, the time for lane change can be obtained by the following formula:

$$\Delta_{t_{c2}-t_{c1}} = f_{t_{c1}}(X(t_{c1}), Y(t_{c1})) \cdot f_{t_{c2}}(X(t_{c2}), Y(t_{c2})). \quad (5)$$

According to the above definition, >0 indicates a left lane change and <0 indicates a right lane change.

Similarly, when there are other vehicles in the target lane, according to the principle of safe lane change, the speed and distance of the lane-change vehicle must exceed those of the vehicle in the target lane. Moreover, the lateral speed of the lane-change vehicle must be greater than that of the vehicle in the target lane, and the lateral distance difference of the lane-change vehicle must be greater than twice the length of the vehicle.

On this basis, lane-change experiments were carried out for two different scenarios where vehicles were present and absent in the target lane, and the lane-change states during a left lane change and a right lane change were compared. Figures 3 and 4, respectively, present the results of left lane change and right lane change with no vehicles in the target lane.

The curves in Figure 3 represent the simulated trajectories obtained through system recognition, for three left lane-change tests. The data labeled by o , x , and $*$ indicate the actual values of these three tests based on edge computing of results acquired and obtained from vehicle-borne unit and roadside units, and their corresponding relationships to simulated trajectory curves are presented in blue, black, and

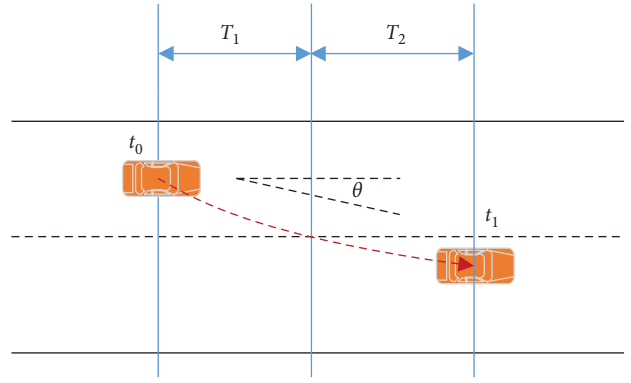


FIGURE 2: Schematic diagram of a lane-change process.

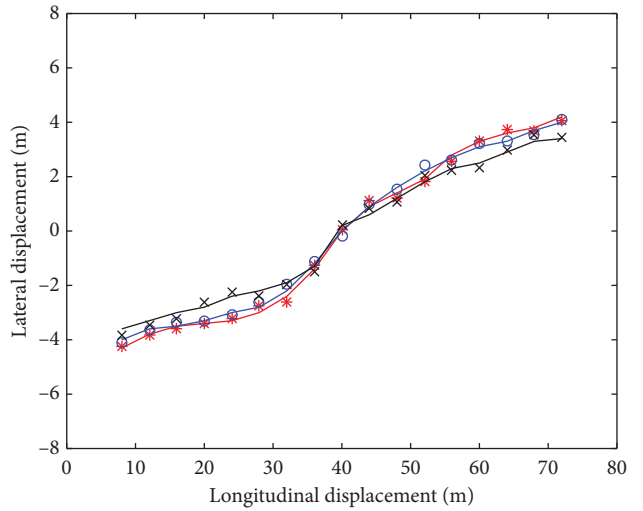


FIGURE 3: Actual test data and simulation data of left lane change.

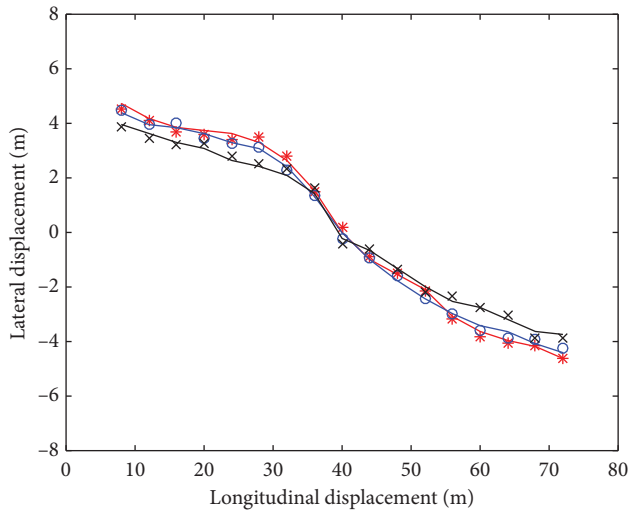


FIGURE 4: Actual test data and simulation data of right lane change.

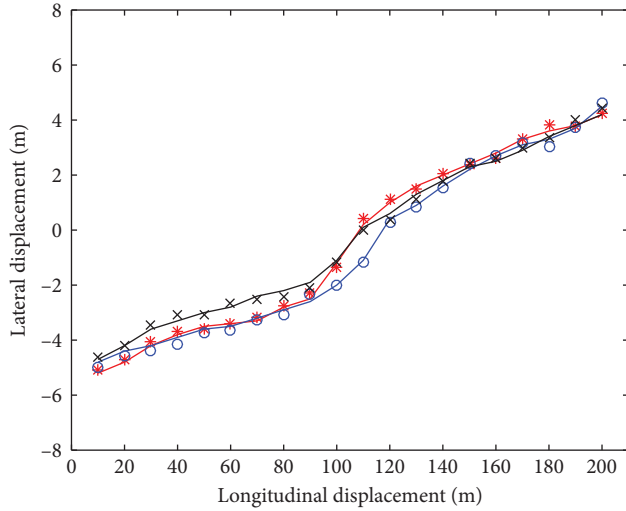


FIGURE 5: Actual test data and simulation data of left lane change.

red, respectively. Figure 4 displays the results for three right lane-change tests. Under standard operating conditions, the standard deviations between the actual value and the simulated trajectory for the three left lane-change operations were 0.78, 0.82, and 0.88, respectively, thus obeying a normal distribution. The lateral displacements in right lane-change operations were greater those in left lane-change operations, and it can be attributed to the seat location of the driver. The standard deviations for the three right lane-change tests were 0.92, 0.93, and 0.89 (slightly large than those of left lane-change operations), thereby still obeying a normal distribution despite slightly larger.

When there were other vehicles in the target lane, tests were carried out in the same way. Figures 5 and 6, respectively, present the results of left lane change and right lane change with vehicles in the target lane and with trajectory recognition based on actual data acquired from the vehicle-borne unit and roadside units. The results are compared with the actual data, as shown in Figure 5.

In comparison to the lane-change test results with no vehicles in the target lane, the longitudinal distance for completing the lane-change operation slightly increased in this scenario. The average longitudinal distance for completing a left lane change increased by 1.42 meters, whereas the average lateral distance increased significantly (nearly 93.89 meters). It happened because the obtained results were affected by an increased lane changing distance, the speed of vehicles in the target lane, and a longer distance to ensure safety.

Under standard operating conditions, the standard deviations between the actual value and the simulated trajectory for the three left lane-change tests were 1.12, 1.21, and 1.19, respectively, thus demonstrating a normal distribution. The lateral distances in right lane-change operations were greater than those in left lane-change operations, and it can be ascribed to the seat position of the driver. The standard deviations, for the three right lane-change tests were 1.36, 1.53, and 1.47, respectively (slightly larger than those of left lane-change operations), thereby still conforming to a normal distribution.

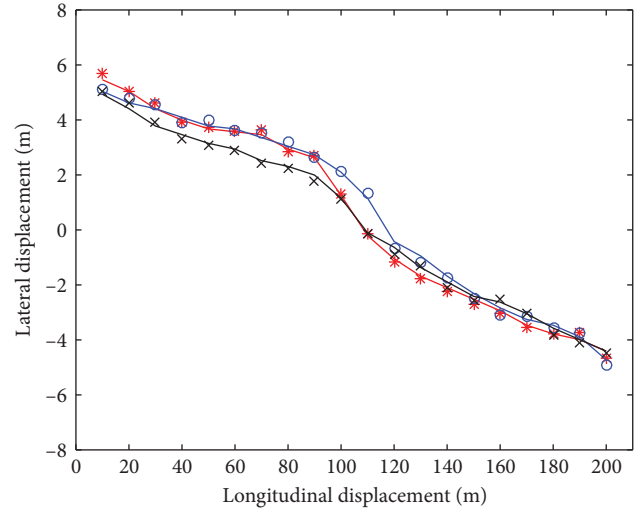


FIGURE 6: Actual test data and simulation data of right lane change.

5. Conclusion

The conventional automobile motion trajectory is mainly constrained by the kinematic and dynamic characteristics and real-time issues of a vehicle. In recent years, the research on smart car movement trajectories has paid more attention to the application of V2X communication among conventional automobiles. Transportation vehicles generally face a highly complex and stochastic driving environment. In order to strengthen the driving safety management, the present paper considered the single-vehicle lane-change process as a research topic. Based on electronic registration identification of motor vehicles and the associated reader/writer, mobile edge computing, and self-organizing feature mapping algorithms, a stochastic parameter recognition method was proposed for the driving trajectory simulation process, and its feasibility was verified through testing. A vehicle trajectory data acquisition method was first proposed based on electronic registration identification of motor vehicles, and the detailed description from the wiring and installation of the readers were then provided to the acquisition, calibration, and application of data. The lane-change process was mathematically modeled by mobile edge computing and self-organizing feature mapping, and the obtained simulation results were compared with actual test data. It was found that the standard deviation between the actual value of the lane-change operation and the simulated trajectory conformed to a normal distribution. Therefore, the proposed method can effectively improve the accuracy of intelligent vehicle trajectory planning and the driving safety during lane-change operation.

Data Availability

The original data used to support the findings of this study are restricted by the relevant law-enforcement departments in order to protect vehicle information privacy and law enforcement basis. Data are available from relevant law-enforcement departments for researchers who meet the criteria for access to confidential data.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jingfeng Yang conceptualized the study, developed methodology, carried out formal analysis, and wrote the original draft. Zhiyong Luo developed methodology and supervised the study. Nanfeng Zhang validated the study and carried out data curation. Jinchao Xiao carried out data curation, formal analysis, and visualization. Honggang Wang validated the study and contributed to resources. Shengpei Zhou wrote, reviewed, and edited the article. Xiaosong Liu contributed to software and resources. Ming Li investigated the study, contributed to funding acquisition, and administered project.

Acknowledgments

This research was funded by the National Key Research and Development Program (no. 2017YFD0700602), 2018 Industrial Internet Innovation and Development Project-Basic Standards and Experimental Verification of Industrial Internet Edge Computing, Key Research and Development Plan of Shaanxi Province (2018ZDXM-GY-041), and Foshan Entrepreneurship and Innovation Team Project (2017IT100032).

References

- [1] J. Yang, Z. Zhang, N. Zhang et al., "Vehicle text data compression and transmission method based on maximum entropy neural network and optimized Huffman encoding algorithms," *Complexity*, vol. 2019, Article ID 8616215, 9 pages, 2019.
- [2] H. Jiang, W. Fang, and J. Huang, "Design of parking management system based on the electronic registration identification," *China Science and Technology Information*, vol. 28, no. 23, pp. 78-79, 2016.
- [3] R. Wu and H. Yang, "Application and research of the electronic registration identification in traffic management field," *Police Technology*, vol. 33, no. 3, pp. 24-26, 2017.
- [4] Y. Tao, X. Yan, T. Wang et al., "Intelligent transportation system development strategy for future intelligent society," *Science & Technology Review*, vol. 37, no. 34, pp. 48-53, 2016.
- [5] W. Lang, S. Tian, Schwarzc et al., "A Review of application of driving simulation technology in R&D of vehicular intelligence technology," *Journal of Highway and Transportation Research and Development*, vol. 34, no. 12, pp. 140-150, 2017.
- [6] C. Zuo, Z. You, and X. Niu, "The application of the electronic registration identification in public security traffic management," *Transportation Enterprise Management*, vol. 32, no. 1, pp. 91-93, 2017.
- [7] Z. L. Sun, "The application and practice of the electronic registration identification," *China Information Security*, vol. 7, no. 10, pp. 68-70, 2016.
- [8] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100-116, 2003.
- [9] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501-525, 2005.
- [10] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263-282, 2002.
- [11] M. Wu, S.-L. Dai, and C. Yang, "Mixed reality enhanced user interactive path planning for omnidirectional mobile robot," *Applied Sciences*, vol. 10, no. 3, p. 1135, 2020.
- [12] J. Luo, Z. Lin, Y. Li, and C. Yang, "A teleoperation framework for mobile robots based on shared control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 377, 2020.
- [13] H. Kong, C. Yang, G. Li, and S.-L. Dai, "A sEMG-based shared control system with No-target obstacle avoidance for omnidirectional mobile robots," *IEEE Access*, vol. 8, no. 1, pp. 26030-26040, 2020.
- [14] D. Grewe, M. Wagner, M. Arumaithurai et al., "Information-centric mobile edge computing for connected vehicle environments: challenges and research directions," in *Proceedings of the Workshop on Mobile Edge Communications*, ACM, New York, NY, USA, August 2017.
- [15] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudan, "Software defined networking-based vehicular ad-hoc network with fog computing," in *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 1202-1207, IEEE, Ottawa, Canada, May 2015.
- [16] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, "Location-aware services over vehicular ad-hoc networks using car-to-car communication," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1590-1602, 2007.
- [17] S. Dombush and A. Joshi, "Street smart traffic: discovering and disseminating automobile congestion using VANETs," in *Proceedings of the 2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*, April 2007.
- [18] Q. Zhang and J. H. Zhao, "A model for automatic collection and dynamic transmission of traffic information based on VANET," in *Proceedings of the 2012 15th International IEEE Conference on Intelligent Transportation Systems*, pp. 373-378, IEEE, Anchorage, AK, USA, September 2012.
- [19] M. Gramaglia, M. Calderon, and C. J. Bernados, "ABEONA monitored traffic: VANET-assisted cooperative traffic congestion forecasting," *IEEE Vehicular Technology Magazine*, vol. 9, no. 2, pp. 50-57, 2014.
- [20] F. Terroso-Saenz, M. Valdes-Vela, C. Sotomayor-Martinez, R. Toledo-Moreo, and A. F. Gomez-Skarmeta, "A cooperative approach to traffic congestion detection with complex event processing and VANET," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 914-929, 2012.
- [21] W. Li, C. Yang, Y. Jiang, X. Liu, and C. Su, "Motion planning for omni-directional wheeled mobile robot by potential field method," *Journal of Advanced Transportation*, vol. 2017, Article ID 4961383, 11 pages, 2017.
- [22] R. Bauza and J. Gozalvez, "Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1295-1307, 2013.
- [23] S. Alexander, "Cluster analysis for large scale gene expression studies," pp. 28-31, Alexander Sturn, Rockville, MD, USA, 2000, M.S. thesis.
- [24] C. Xu, Y. Yang, and W. Chen, "Application and analysis of self-organizing feature map," *Journal of ChangChun Teachers College (Natural Science)*, vol. 24, no. 4, pp. 55-58, 2005.

- [25] C. Li, L. Ning, and P. Shi, "The principle and application of self-organizing mapping network in the cluster of cultivated and use pressure in China," *Journal of Beijing Normal University(Natural Science)*, vol. 51, no. 5, pp. 543–547, 2006.
- [26] F. Haitao, "The estimation method of parameter identification for aircraft stochastic control system," M.S. thesis, Shenyang Ligong University, Shenyang, China, 2015.
- [27] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 3, pp. 777–787, 2019.
- [28] H. Wang, S. Wang, J. Yao et al., "Effective anti-collision algorithms for RFID robots system," *Assembly Automation*, vol. 40, no. 1, pp. 55–64, 2019.