

Received April 8, 2020, accepted May 3, 2020, date of publication May 8, 2020, date of current version June 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993462

PCM: A Positioning Calibration Mechanism for Connected Autonomous Vehicles

YUHAO LING^{1,2,3}, XINGHE CHU^{1,2,3}, ZHAOMING LU^{1,2,3}, LUHAN WANG^{1,2,3,4}, AND XIANGMING WEN^{1,2,3}

¹Beijing Laboratory of Advanced Information Networks, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China

²Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China

³School of Information and Communication Engineering, Beijing University of Posts and Telecommunications (BUPT), Beijing 100876, China

⁴Witcomm Open Source Communication Research Institution, Beijing 100876, China

Corresponding author: Zhaoming Lu (lzy0372@bupt.edu.cn)

This work was supported by the National Science and Technology Major Project under Grant 2018ZX03001018. Thank the IEEE 5G Automated Driving Innovation and Application Demonstration Center for providing a testing ground for this paper.

ABSTRACT Connected autonomous driving is considered as a potential way to improve traffic safety. However, when positioning information is delivered through network, the network time delay could result in additional positioning errors, which has been proved by the experiment done in this paper in XuChang automatic driving center that the influence of one-way time delay will lead to meter-scale positioning error. In this paper, we propose a positioning calibration mechanism for connected autonomous vehicles called PCM. PCM firstly estimates the network time delay and then saves the calibrated location information in the MEC server to reduce the positioning error caused by network time delay. Experiments in FangShan and XuChang automatic driving centers verify the performance of PCM, and the result shows that PCM can reduce the positioning error to centimeter-level. For example, under 100ms time delay, the average positioning error without PCM is 119.67cm, while the average positioning error with PCM is 3.03cm. Experimental results show that PCM can effectively reduce the positioning error caused by a one-way time delay to centimeter-level, which is strictly required for positioning autonomous vehicles in automatic driving.

INDEX TERMS Automatic driving, high precision positioning, one-way time delay, the transformation of a coordinate system, mobile edge computing.

I. INTRODUCTION

The idea of automatic driving has been grown for a long time. The first widely recognized autonomous vehicle “Stanford Cart” [1] was born in 1961. More and more autonomous vehicles are gradually entering the market nowadays. The positioning has always been an essential issue for automatic driving [2]. There are many applications of high precision positioning in the automatic driving system, such as path planning and obstacle avoidance, which substantially improve the accuracy and safety of autonomous vehicles [3].

At present, the most famous positioning technology for autonomous vehicles is Real-time Kinematic (RTK) technology [4]. It is based on Global Positioning System (GPS), by which the onboard satellite receiver provides centimeter-level accurate position information to autonomous vehicles. However, GPS signals may be blocked by buildings or vegetation in densely populated urban areas, causing a sharp decline in positioning accuracy. So a solution named Fusion

Positioning [5]–[7] comes forth, that is to coordinate position information of GPS with other onboard sensors. Usually, the Inertial Measurement Unit (IMU) is needed to provide motion information (Speed, acceleration, direction angle, pitch angle, etc.) for vehicles in real-time. This IMU-based integrated navigation system enables vehicles to maintain centimeter-level positioning accuracy when losing GPS signals within a short distance [8]. However, there still exist accumulated errors.

In automatic driving scene, vehicles need to obtain not only their own accurate position information but also position information of other surrounding vehicles to furtherly assure safety [9]. In response to this, autonomous vehicles should be equipped with laser radar, millimeter-wave radar, binocular camera, and other sensing devices to detect surroundings, yet such devices are vulnerable to environmental interference, which could result in enormous error [10], [11]. The key to solving this problem is to connect autonomous vehicles to the network and introduce Road Side Unit (RSU) to automatic driving roads [12], [13]. The RSU is static and can provide additional positioning information for vehicles under extreme

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Yang¹.

conditions. All RSUs and vehicles upload their perceived environmental information and high-precision positioning information to the Mobile Edge Computing (MEC) server to build a high-definition (HD) map [14]. So the MEC server can update these messages in corresponding areas managed by itself in real-time. Especially when one vehicle's onboard equipment is disturbed by environmental interference and then causes errors as mentioned before, current environmental information measured by other surrounding vehicles still can be sent to this influenced vehicle for navigation through MEC [15]. In brief, MEC servers integrate multivehicle information with RSUs to provide more accurate and stable positioning services for interconnected vehicles.

But there existing a one-way delay when either RSUs send the measured information or vehicles send their own positioning information to the MEC server, as shown in FIGURE 1. For example, if the one-way delay is 10ms, which is a typical end-to-end delay of 5G [16], positioning error caused by this delay could be 16.7cm when the vehicle reaches 60km/h speed. It means that by the time the MEC server receives positioning information uploaded by the vehicle, it has already moved 16.7cm forward. However, automatic driving requires the positioning accuracy of vehicles to be within 10 cm. Moreover, it would be larger under the high-speed driving scene, or when the network environment is unstable. In a word, positioning error caused by neglect of time delay may result in serious safety problems.

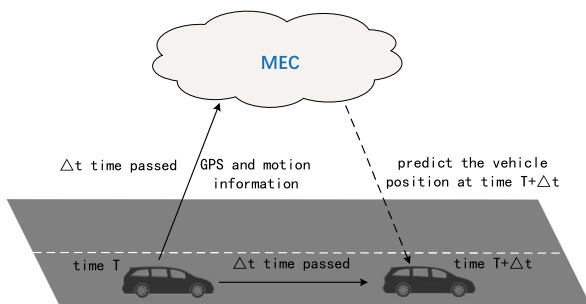


FIGURE 1. The influence of time delay on the location information in the connected automatic driving architecture.

In order to solve this problem, a collaborative driving system [17] is designed from the perspective of joint network and control, so that the worst-case minimum safe distance affected by one-way time delay is determined. On the other hand, traffic efficiency may be reduced because of the larger distance kept between vehicles. So the 5G network is introduced to speed up data transmission, but it is still essential to establish a mechanism to correct the positioning error caused by network time delay.

The main contributions of this paper are as follows:

1. We design a method to calculate the one-way delay existing in the transmission of vehicles and MEC data by using the GPS satellite timing model. Furtherly, we propose a positioning calibration mechanism named PCM to predict real-time vehicle position via the calculation of one-way time delay mentioned above.

2. We establish two scenarios to simulate the communication state between vehicles and the MEC server in the process of automatic driving and verify the feasibility of PCM in these two practical scenarios.

3. We analyze the predicted positioning error of PCM in the actual scene.

The remainder of this article is organized as follows. Section II introduces the concrete realization principles of our algorithm. Section III shows the structure of the whole testing system and preparatory work before the experiment. Section IV demonstrates the performance of this algorithm in two practical scenarios and then compares the MEC positioning error with or without algorithm. In the end, Section V summarizes this article.

II. POSITIONING CALIBRATION MECHANISM

PCM algorithm consumes less computing resources and can be deployed on the vehicle's on-board controller and the MEC server. The PCM deployed on the MEC server can calibrate the vehicle's position errors caused by information propagation delays. The flow of the PCM algorithm is shown in FIGURE 2. The vehicle sends its state information as a message to the MEC server, which contains sending time, vehicle position, direction angle, pitch angle, velocity, and acceleration. After receiving the message from the vehicle, the MEC server calculates the one-way time delay of message propagation. Then MEC can use the vehicle's state information and one-way time delay to calculate the exact location of the vehicle at the moment (when the message is received). The core of PCM algorithm lies in three aspects:

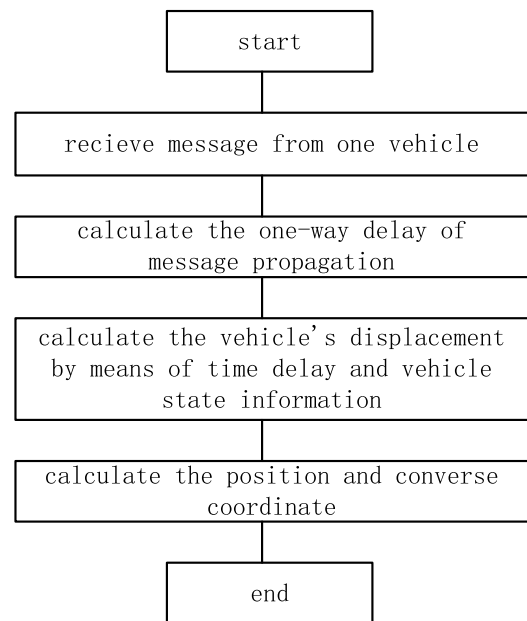


FIGURE 2. A flowchart of the PCM algorithm running on the MEC server.

1. How to calculate one-way time delay accurately.
2. Establish a connection between a cartesian coordinate system with the vehicle as to the origin and a GPS coordinate system with the earth's center of mass as the origin.

3. How to calculate the position of the vehicle by using its state information and one-way time delay.

A. ONE-WAY TIME DELAY

When the MEC server receives the message from the vehicle controller, it can't store the vehicle's location information into the database immediately. There is a time delay in the transmission of the message, so the position of the vehicle at the time when the MEC server receives the message is different from it at the time when the message is sent by the vehicle controller. The key to eliminating the position deviation is to know the specific value of the one-way time delay of the data transmission. So, if the time of the vehicle controller and the MEC server has been strictly synchronized, we can calculate the one-way time delay directly:

$$\Delta t = t1 - t2 \quad (1)$$

where Δt is the one-way time delay of data transmission. $t1$ is the message's receiving timestamp obtained from the MEC server, and $t2$ is the message's sending timestamp that the MEC server gets from the message. The vehicle's onboard controller puts $t2$ in this message.

B. COORDINATE TRANSFORMATION

At the beginning of section II, we know that PCM predicts the position of the vehicle by using its direction angle, pitch angle, and GPS position. Direction angle and pitch angle are obtained by using the vehicle as to the reference frame, which is a space rectangular coordinate system. So we need to convert the GPS coordinate system to space rectangular coordinate system to build the connection between direction angle, pitch angle, and GPS position.

World Geodetic System 1984 (WGS84) is a coordinate system established for deeply using GPS global positioning system [18], which is a kind of geocentric coordinate system adopted internationally, also known as Earth-Centered and Earth-Fixed (ECEF) coordinate system. That means its origin is the center of mass of the earth, and its Z-axis points to the direction of the agreed earth pole (North Pole), which is defined by BIH (Bureau International de l'Heure). Its X-axis always points to the intersection of the equator and the meridional plane of Greenwich, and Y-axis meets the right-handed rule with the X-axis and Z-axis in the equatorial plane. Its X-axis and Y-axis rotate with the earth's rotation, but the whole coordinate system is stationary with respect to the earth. The latitude and longitude given by GPS is a special coordinate established in the WGS84 coordinate system. It can be transformed with space rectangular coordinate system.

The GPS uses latitude and longitude to position the vehicle. But the direction angle and pitch angle can not be directly calculated with the longitude and latitude coordinates. They base on another coordinate system that uses the vehicle as to the reference point. When the MEC server processes the data from the vehicle controller, it must convert the latitude and longitude coordinates to space rectangular coordinate system

in WGS84 [19]. So the current position of the car can be represented by a coordinate value on the WGS84 XYZ-axis. Then we establish the transformation relationship between the vehicle coordinate system and the wgs84 coordinate system.

1) CHANGE GPS COORDINATE TO SPACE RECTANGULAR COORDINATE

According to the definition of latitude and longitude coordinates and the relevant parameters defined by WGS84 [19]. We can write the following coordinate transformation formula:

$$X = (N + H) \cdot \cos B \cdot \cos L \quad (2)$$

$$Y = (N + H) \cdot \cos B \cdot \sin L \quad (3)$$

$$Z = \left[N \cdot (1 - e^2) + H \right] \cdot \sin B \quad (4)$$

where X, Y, Z is the value of the X-axis, Y-axis, and Z-axis in the WGS84 space rectangular coordinate system. B is latitude, L is longitude, and H is altitude. The MEC server can obtain B, L , and H from the data sent by the vehicle controller.

According to the earth measurement parameters defined by the WGS84 coordinate system, the long radius $a = 6378160\text{m}$ and the short radius $b = 6356775\text{m}$.

The first eccentricity e of earth:

$$e^2 = (a^2 - b^2) / a^2 \quad (5)$$

So we can calculate the value of the first eccentricity of the earth $e = 0.0818196$.

N is the radius of the curvature of the earth at the current position of the car, and the value of N is related to latitude.

Define W as the first auxiliary coefficient:

$$W = \sqrt{1 - e^2 \cdot \sin^2 B} \quad (6)$$

The value of N can be expressed as:

$$N = a / W \quad (7)$$

Therefore, when we know the specific position of the vehicle, we can calculate the current value of the radius of curvature N according to the latitude B . We can figure out the existing space rectangular coordinates of the vehicle by substituting N in formula (2), (3), and (4).

2) CONVERT SPACE RECTANGULAR COORDINATE BACK TO GPS COORDINATE

When we complete the calculation by using a space rectangular coordinate system, we need to convert the rectangular coordinate of the vehicle to the longitude and latitude coordinate in the WGS84 coordinate system. Only GPS coordinates can be used to locate vehicles in the database of the MEC server.

Suppose the real space rectangular coordinate of the car after the prediction is (X, Y, Z) . According to the previous formula (2), (3) and (4).

(2) divided by (3) has the following formula:

$$L = \arctan(Y/X) \tag{8}$$

So the longitude L can be easily calculated.

(4) divided by (3) has the following formula:

$$\left[1 - \frac{N \cdot e^2}{N + H} \right] \cdot \tan B = \frac{(z \cdot \cos L)}{X} \tag{9}$$

So we can get $\tan B$ as:

$$\tan B = \frac{z \cdot \cos L}{X \cdot \left(1 - \frac{N \cdot e^2}{N + H} \right)} \tag{10}$$

It is very complicated to solve this equation directly. We want to find latitude B . But in formula (6) and formula (7), curvature N is related to B . So the idea of the iterative solution is adopted here. The latitude B , longitude L , and altitude H change very little over a period of time when driving automatic vehicles. So the change in N is also tiny. The first approximate value $B1$ of latitude B can be solved by using the previous H and N values.

The current relatively accurate curvature $N1$ is calculated from formula (6) and formula (7), and the current altitude approximation $H1$ can be obtained through formula (2).

Then continue to substitute $H1$ and $N1$ into formula (10) to solve the more accurate latitude value $B2$. Through $B2$, a more accurate value of elevation $H2$ and curvature $N2$ can be obtained.

Through multiple iterations, the final result Bn will be very close to the exact value of latitude B . Automatic driving requires the errors of latitude B , and altitude H are eight decimal places. Through the actual test, the accuracy of using two iterations has been able to meet the calculation requirements.

At this point, we can obtain the predicted GPS coordinates of the car's current position.

C. PREDICT THE VEHICLE'S CURRENT POSITION

The GPS coordinates describe the vehicle's position information, while direction angle and pitch angle describe the vehicle's direction information. Direction angle refers to the angle between the current direction of the vehicle and the real north direction (tangent direction of the true meridian of the earth), which is 0° (360°) to the north and 90° to the east. Pitch angle refers to the angle between the direction of the vehicle and the current horizontal plane. If it is above the 0 scale, it means that the pitch angle is positive and vice versa.

We think that the motion angles of the vehicle keep the same for a very short time. So during the one-way time delay of the transmission of data from the vehicle controller to the MEC server, the vehicle keeps moving in a straight line along the current angle.

We use the unit direction vector \vec{m} to represent the vehicle's motion angle in the WGS84 space rectangular coordinate. We need to find the relationship between \vec{m} and the car's direction angle and pitch angle, and figure out the unit direction vector \vec{m} .

The magnitude of the unit vector \vec{m} is 1:

$$|\vec{m}| = 1 \tag{11}$$

Define the direction angle as θ and the pitch angle as φ . We can draw a vector figure of the vehicle moving relative to the earth.

As shown in FIGURE 3, according to the WGS84 space rectangular coordinate system model, point O is the center of mass of the earth, and the Z -axis passes through the North Pole. We define the current position of the vehicle as point P , and let the vector through the point O and point P be vector \vec{r} . The horizontal plane is the tangent plane of the earth at the point P , so that the vector \vec{r} is a normal vector of the horizontal plane. The intersection of the Z -axis and the horizontal plane is point Q , and vector \vec{PQ} is the real north vector (define as vector \vec{N}) of the car.

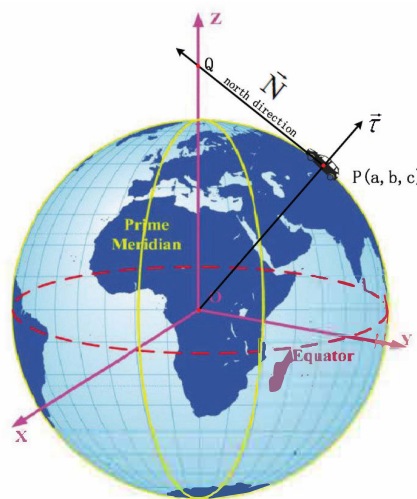


FIGURE 3. Vector figure of the vehicle moving relative to the earth.

Then we take the car as the origin and construct another space rectangular coordinate system, as shown in FIGURE 4.

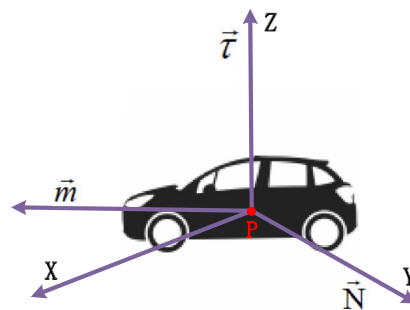


FIGURE 4. Space rectangular coordinate system of vehicle.

Point P is the current position of the car and the origin of this new coordinate system. In other words, Point P is the center of mass and the positioning center of the vehicle. Y -axis is the real north direction of the current position (vector \vec{N} in FIGURE 3). The x -axis is perpendicular to Y -axis, and the plane XPY is parallel to the horizontal plane. So The Z -axis is

the normal vector $\vec{\tau}$ of the horizontal plane, and the direction vector of the vehicle is vector \vec{m} .

The direction angle θ is the angle between vector \vec{m} and the real north direction, and the pitch angle φ is the angle between \vec{m} and the horizontal plane. So combined with the coordinate system in FIGURE 3 and the definition of direction angle and pitch angle, the direction angle θ is the angle between vector \vec{m} and plane ZPY , the pitch angle φ is the angle between vector \vec{m} and plane XPY .

Define the projection of vector \vec{m} on plane XPY as vector \vec{w} . So the direction angle θ is the angle between vector \vec{w} and vector \vec{N} , and the pitch angle φ is the angle between vector \vec{w} and vector \vec{m} .

The Z-axis, \vec{m} and projection vector \vec{w} are in the same plane, so:

$$\cos\langle\vec{\tau}, \vec{m}\rangle = \sin\langle\vec{m}, \vec{w}\rangle = \sin\varphi \quad (12)$$

The north vector \vec{N} of the vehicle is only related to the vehicle's current position, so \vec{N} can be calculated by GPS coordinates. According to the Tri Cosine Theorem, we can write:

$$\cos\langle\vec{m}, \vec{N}\rangle = \cos\langle\vec{m}, \vec{w}\rangle \cdot \cos\langle\vec{w}, \vec{N}\rangle \quad (13)$$

$$\cos\langle\vec{m}, \vec{N}\rangle = \cos\varphi \cdot \cos\theta = \frac{\vec{m} \cdot \vec{N}}{|\vec{m}| \cdot |\vec{N}|} \quad (14)$$

The unit direction vector \vec{m} of the vehicle can be obtained by solving the equations (11), (12) and (14).

Because cosine is an even function, vector \vec{m} will have two solutions $\vec{m}1$ and vector $\vec{m}2$. These two solutions $\vec{m}1$ and $\vec{m}2$ are symmetric with respect to the north plane ZPY .

To solve this problem, we need to know the approximate value \vec{m}' of the direction vector \vec{m} . It is a continuous process for the vehicle controller to send data to the MEC server, and the change of vector \vec{m} is smooth. Therefore, we can know the position \vec{P}_{last} of the car at the previous data receiving time. The approximate value \vec{m}' of vector \vec{m} can be obtained:

$$\vec{m}' = \vec{P}_{now} - \vec{P}_{last} \quad (15)$$

Obviously \vec{P}_{now} and \vec{P}_{last} are space rectangular coordinate after processing.

Find the cosine similarity between \vec{m}' and solution $\vec{m}1$ or $\vec{m}2$ respectively, and judge which is the real solution according to the cosine similarity. The solution with large cosine similarity is the real direction vector \vec{m} .

The one-way time delay Δt is tiny, so we can approximate the vehicle's motion model in Δt as a linear motion model with uniform acceleration. This model is more accurate when Δt is smaller. According to the equation of uniformly accelerated motion:

$$s = \int_{\Delta t} v(t) \cdot dt \quad (16)$$

We can figure out the displacement of the vehicle along the current direction vector \vec{m} . Suppose the space rectangular

coordinate of the car at point P is (x, y, z) . \vec{m} is a unit direction vector. Suppose \vec{m} is (a, b, c) , so the coordinate of the car after the one-way time delay Δt is $(x + as, y + bs, z + cs)$.

So we can calculate the current position of the vehicle after a one-way time delay.

III. SYSTEM ARCHITECTURE AND PREPARATION

In this experiment, we adopt the Integrated Navigation System (INS) [20], which combines IMU and GPS to receive vehicle state information and positioning information. The INS sends data to the vehicle controller. When the controller receives the data sent by INS, it will add the current timestamp before the data and send the processed message to the MEC server through the network. When the MEC server receives this message, it records receive timestamp. The one-way time delay of data transmission is obtained by subtracting the timestamp of sending moment from that of the receiving moment. Then the position of the vehicle in this receiving moment can be calculated by using the state information in this message.

FIGURE 5 shows the on-board equipment used in this experiment.



FIGURE 5. The automatic driving terminal designed by our laboratory, which combines INS, vehicle controller and Customer Premise Equipment (CPE). Internal structure: 1. Miniature transformer. 2. GPS module(In the actual test we replaced it with INS). 3. CPE. 4. Vehicle controller (Raspberry Pi 3).

A. TIME SYNCHRONIZATION

If the timestamp of the vehicle controller is not synchronized with this of the MEC server, we cannot obtain the one-way time delay of data transmission by subtraction directly. Therefore, it is necessary to keep the vehicle controller timestamp highly synchronized with the server timestamp. The general solution to solve this time synchronization problem is using

Network Time Protocol (NTP) [21]. But NTP requires the communication process between the vehicle controller and MEC server keeping a stable time delay, especially the uplink and downlink time delay must be consistent. The vehicle controller connects to the MEC server through the LTE network, which cannot guarantee the same time delay in each data transmission process and the same uplink and downlink time delay in one data transmission process. Therefore, the vehicle controller can't synchronize with the MEC server by the NTP protocol.

GPS is a global satellite radio navigation system, which can provide continuous, realtime positioning services for many users. At the same time, GPS navigation messages contain accurate time [22]. GPS timing system [23] uses a receiver to receive the time signal of the "atomic clock" from satellite to obtain GPS time. GPS timing system can ensure the accuracy of the clock by receiving at least one GPS satellite signal so that other devices can be accurately synchronized.

The easiest way for the vehicle controller to get accurate timestamp is to synchronize the PPS (Pulse Per Second) signal from a GPS time module, which is cheap but the accuracy can reach 100ns [24]. We connected the vehicle controller and the MEC server with two GPS time module separately to synchronize their timestamp. This test uses millisecond timestamp, so the accuracy of 100ms can fully satisfy the test requirements.

FIGURE 6 shows the device and GPS time model connection architecture of this experiment.

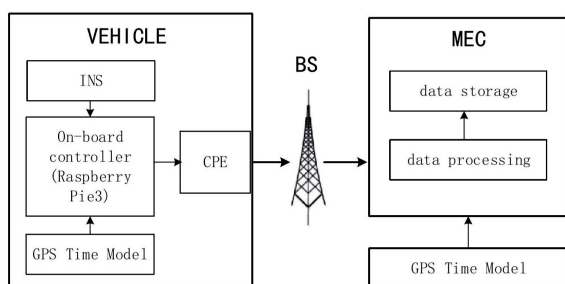


FIGURE 6. System architecture.

B. VEHICLE STATUS DATA COLLECTION

INS can output at most 100Hz positioning information and state information of the vehicle. The accuracy of positioning is centimeter-level. The positioning information includes longitude, latitude, and altitude. The state information includes speed, acceleration, direction angle, and pitch angle.

As shown in FIGURE 6, the vehicle controller adopts Raspberry Pi 3 [25], and communicates with INS through its serial port. This controller can add sending timestamp in front of the message after receiving it from INS, and send the message to MEC server by LTE network. Raspberry Pi 3 takes less than one millisecond to process data [26] so that processing delays can be ignored.

IV. ANALYSIS OF TEST RESULTS

The experiment is done in two automatic driving centers to test the performance of PCM.

FIGURE 7 shows the automatic driving experiment center located in Xuchang city, Henan province. The vehicle is connected to the LTE network through on-board CPE. The position of the vehicle is get from INS equipped on the vehicle, which is sent to its remote server located in Beijing. The server in Beijing has a large time delay, so it can better test the accuracy of prediction methods under a large time delay.



FIGURE 7. Xuchang automatic driving experiment center.

FIGURE 8 shows the automatic driving experiment center located in Fangshan District, Beijing. The difference between the Xuchang experiment center and this experiment center is that we use the MEC server here. The MEC server is on the edge of the network, so the delay is generally small and stable.



FIGURE 8. Fangshan automatic driving experiment center.

A. THE EXPERIMENT IN XUCHANG: PERFORMANCE OF THE ALGORITHM IN REAL SITUATIONS

We test the PCM algorithm in Xuchang automatic driving experiment center. The data transmission between the vehicle controller and the remote server is completed by UDP. INS can output vehicle state information at up to 100Hz. However, due to the impact of single transmission packet size, packets loss occurs when TCP transmission frequency is over 10Hz in the actual scene [27], [28]. Therefore, we choose to use UDP to send the message. Packets loss doesn't occur in UDP transmission. But when the data transmission frequency

exceeds 100Hz, sticky packets occurs when the remote server receives a message. It means that the message sent at different times arrives at the same time.

In the actual scene, if the message is sent at the frequency of 50Hz, the sticky packets rate is 15.575%. The occurrence of sticky packets does not affect the prediction accuracy of the PCM algorithm, because as long as the message sending time and receiving time are determined, the exact position of the vehicle at the current time can be predicted. However, it will affect the efficiency of the MEC to collect information, because there's only one piece of message can be used for prediction at one time. Many network resources are wasted when there is a lot of vehicles. Therefore, to save resources and improve the experimental effect, we have used UDP at the frequency of 20Hz to transmit data.

Firstly, the vehicle controller and remote server need to synchronize timestamp through the GPS time module. INS sends positioning and status information to the vehicle controller at the frequency of 20Hz. The vehicle controller adds the current millisecond timestamp before the message and then sends the message to the remote server. When the server receives this message, it will get the receiving timestamp of this message from itself, and then gets the sending timestamp from the message. The one-way time delay of data transmission can be obtained by subtracting these two timestamps, and the current position of this autonomous vehicle can be calculated by the PCM algorithm. Then the remote server stores all messages and results in local files.

To test the accuracy of the PCM algorithm in this scene, we run another program on the vehicle controller at the same time, which collects the vehicle position data from another INS at the frequency of 100Hz.

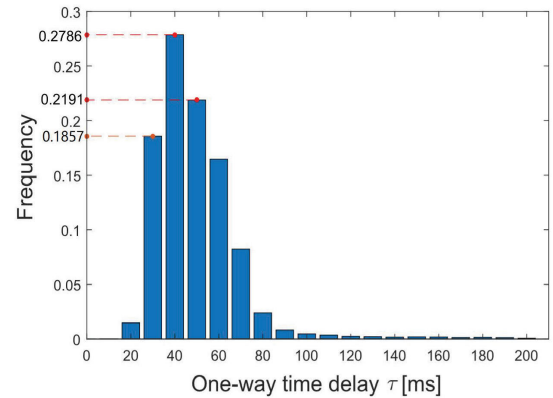
The positioning data collected by the remote server is discrete, but the motion of the vehicle is linear. The solution to this problem is to use interpolation to make the discrete trajectory of the vehicle linear [29] so that we can get the approximate position of the car at a certain time. The exact method of analyzing PCM algorithm errors will be put in the next section.

We make the autonomous vehicle drive 50 laps around the XuChang test site and collect one million pieces of data.

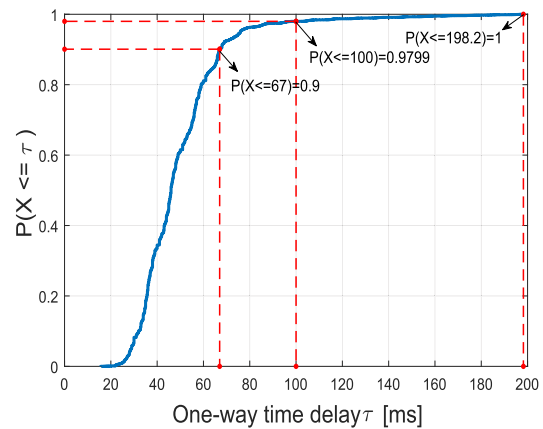
From the measurement principle of PCM in section II, PCM predicts the position of the vehicle based on one-way time delay and the vehicle's current speed. So we suppose that the measurement error is related to these two parameters.

Firstly we analyze the one-way time delay. We extract the one-way time delay of the whole data set and plot the frequency distribution histogram and cumulative distribution function (CDF) plot of the one-way time delay of the autonomous vehicle shown in FIGURE 9.

FIGURE 9 shows the highest one-way time delay is 198.2ms, and the lowest is 15.3ms. The majority of one-way time delay is distributed in the range of 20ms to 80ms, with 96.3% of it less than 80ms and 98% of it less than 100ms. We can calculate the average one-way time delay of the whole data set is 49.4ms.



(a) The frequency distribution histogram of one-way time delay



(b) The CDF of one-way time delay

FIGURE 9. Distribution property of one-way time delay.

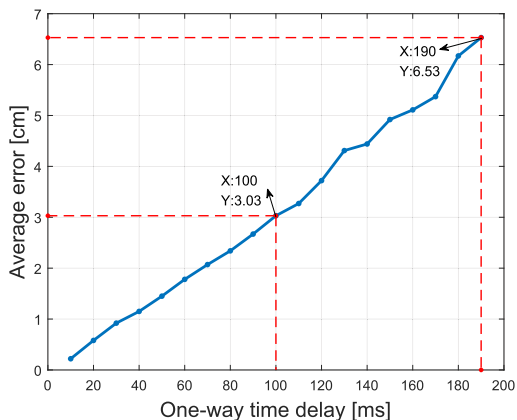
We plot the average positioning error and one-way time delay figure at a resolution of 10ms, which is shown in FIGURE 10(a). Such as when the one-way time delay is 10ms, the figure shows the average error among 5ms and 15ms.

The average error of the PCM algorithm is 1.58cm. The smaller the one-way time delay, the lower the average error of the PCM algorithm. The error is approximately linear with the one-way time delay. When the one-way delay is between 190ms and 200ms, the average error of PCM is 6.132cm, which can reach the centimeter-level positioning accuracy required by automatic driving.

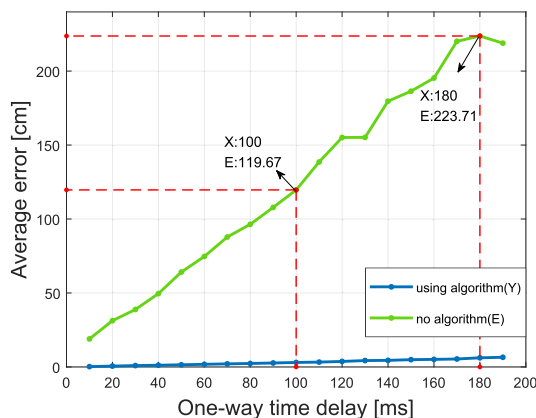
We draw a relationship plot between vehicle positioning error and one-way time delay in FIGURE 10(b) when the PCM algorithm is not used. The positioning error is much higher than 10cm without the PCM algorithm. PCM algorithm significantly improves the accuracy of the remote server to position the vehicle.

We use the same way to plot the frequency distribution histogram and CDF plot of the speed of the vehicle as shown in FIGURE 11.

When the speed is 0, the car is waiting for the traffic light. We drive in actual urban roads, so the speed of the vehicle



(a) Average positioning error of PCM at different time delays



(b) Compare with or without algorithm positioning error

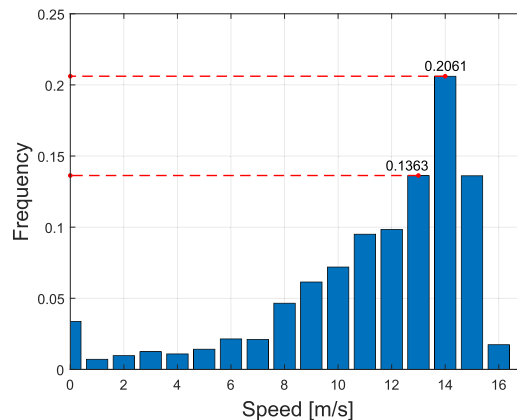
FIGURE 10. Relationship between one-way time delay and positioning error.

is mainly concentrated on 8m/s to 15m/s. The speed data was equally divided into nine intervals according to the interval of the speed of 2m/s. We draw a line chart of the vehicle speed and average error, which is shown in FIGURE 12(a).

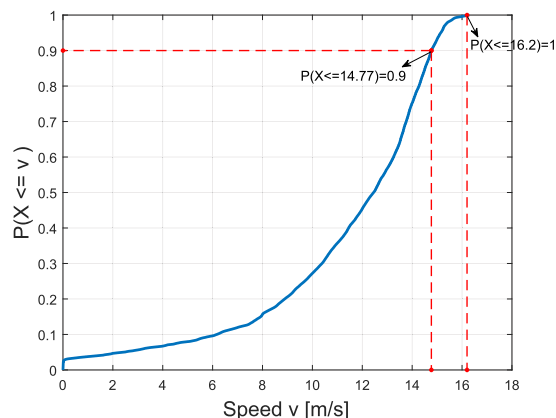
As we can see, the speed of the vehicle is approximately proportional to PCM error. When the speed is 15.5m/s, the max average error is 2.602cm. FIGURE 12(b) shows the relationship between speed and average error without the PCM algorithm. When the speed is 15.5m/s, the max average error without PCM is 111.3cm. It can also be seen that PCM can effectively improve the positioning accuracy.

The frequency distribution histogram and CDF plot of the error data set are shown in FIGURE 13. The maximum error is about 7cm. And about 96% of the error data is less than 3 centimeters, 99% of the error data is less than 5 centimeters. All error data meet the centimeter-level accuracy required by automatic driving.

The experiment in XuChang proves that in the actual automatic driving scene, even if the network environment is not ideal so that the time delay is large, the PCM algorithm can still improve the positioning accuracy to the centimeter-level required by automatic driving.



(a) The frequency distribution histogram of speed



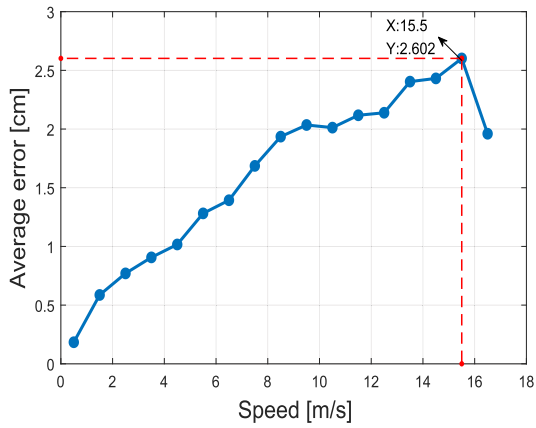
(b) The CDF of speed

FIGURE 11. Distribution property of speed.

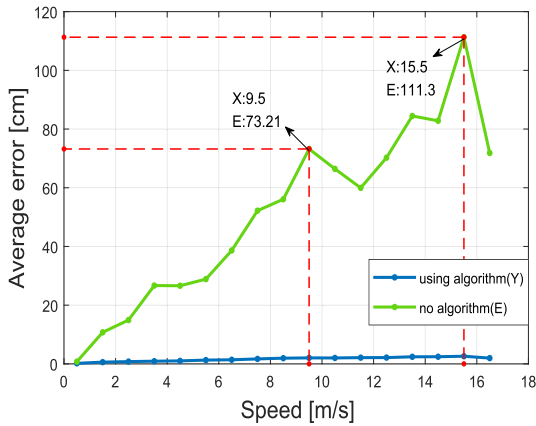
B. THE EXPERIMENT IN FANGSHAN: CALCULATE THE ALGORITHM ERROR ACCURATELY

The network structure in the FangShan test field is different from that in the XuChang test field. In this scene, vehicle data collection and one-way time delay measurement are completed separately. In other words, the vehicle data collection is completed by the vehicle controller itself (to collect high-frequency data), while the vehicle controller and MEC server jointly complete the one-way time delay measurement. The message of the vehicle is not sent during the time delay measurement so that the INS data of the vehicle can be collected at high frequency.

To calculate the error of the PCM algorithm accurately, we use the control variable method. After collecting a large amount of one-way time delay data of a fixed time, we assumed that the time-delay was a fixed value and compared the vehicle position predicted by the PCM algorithm using a fixed time delay with this collected by the actual sensor. We can obtain the accuracy of the algorithm at different time delays. The vehicle controller collects vehicle state data at 50Hz frequency (a fixed interval of 20ms), and the vehicle drives around the test field 50 times, collecting about one million pieces of data.



(a) Average positioning error of PCM with different speed



(b) Compare with or without algorithm positioning error

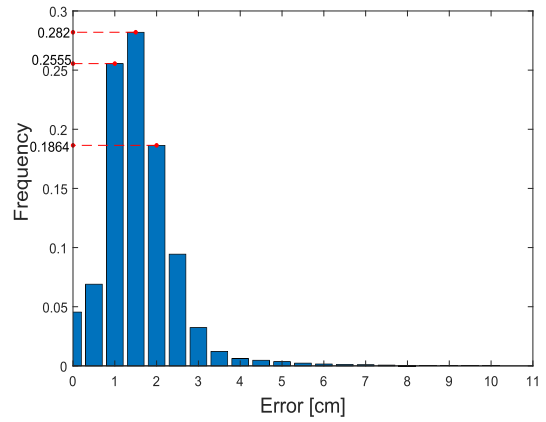
FIGURE 12. Relationship between speed and positioning error.

For example, we have the real data of the vehicle position at the timestamp of 000000030, 000000050, and 000000070, which are tested by INS. We use the real position data at 000000030 and assume the one-way time delay is 20ms, and then calculate the predicted position value of the car at the moment 000000050. By comparing the predicted value with the real position data at 000000050, we can get a more specific algorithm error. Similarly, if the one-way time delay is assumed to be 40ms, the specific error of the algorithm at 40ms can be tested by comparing the real and predicted values at the moment 000000070.

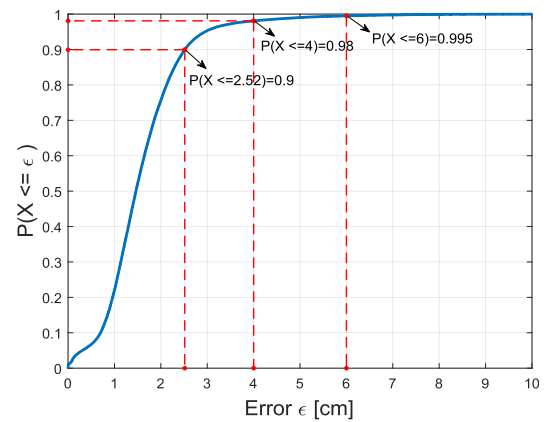
Through this method, we calculated the average error of the PCM algorithm under different one-way time delay and drew the relationship between error and delay in FIGURE 14.

As can be seen from FIGURE 14, the average error is 0.8cm when the time delay is 20ms, and then the error gradually increases linearly and strictly with the increase of time delay. The average error has reached 5.09cm when the time delay is 200ms.

In the future 5G network, the typical end-to-end delay is 5-10ms. In existing 4G networks, the typical end-to-end delay is 50-100ms [30]. We draw a CDF plot of PCM algorithm errors with different typical time delays in FIGURE 15.



(a) The frequency distribution histogram of the whole algorithm position error



(b) The CDF of the algorithm position error

FIGURE 13. Distribution property of the algorithm position error.

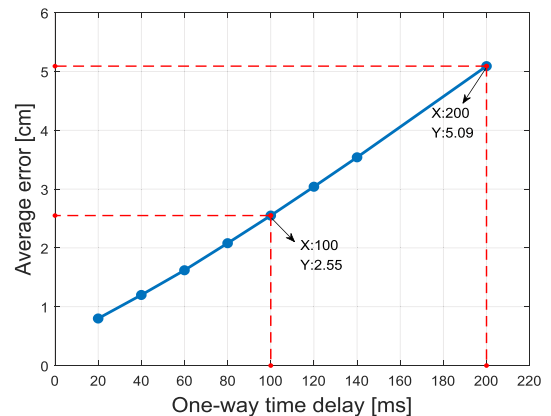


FIGURE 14. Diagram of one-way time delay and PCM positioning error.

The one-way time delay from vehicle controllers to the MEC edge server is about 20ms in the FangShan test field. So the PCM algorithm has high accuracy in these typical time delays and meets the requirements of automatic driving.

We select the maximum typical time delay (100ms) of 4G and plot the frequency distribution histogram of its error data in FIGURE 16. The distribution of the algorithm error is approximately normal distribution, and the majority of

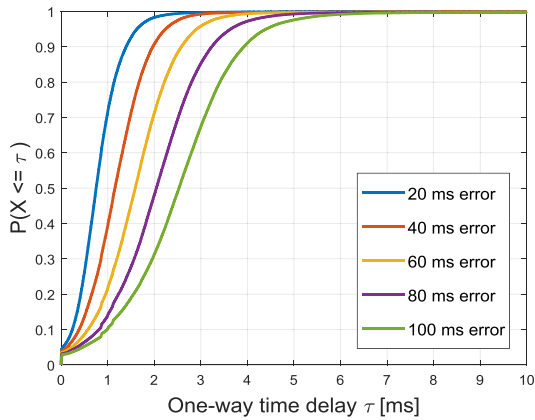


FIGURE 15. CDF of PCM positioning error under different one-way time delays.

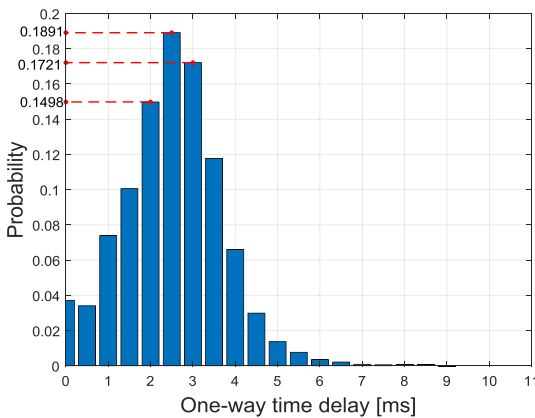


FIGURE 16. The frequency distribution histogram of PCM positioning error when time delay is 100ms.

the error is within the range of 1ms-4ms. The max error is 7.11cm.

We calculate the average error of 100ms time delay under different speed, which is shown in FIGURE 17.

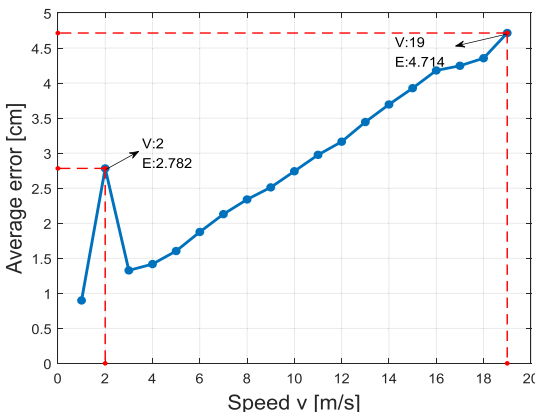


FIGURE 17. Diagram of speed and average algorithm positioning error of 100ms time delay.

In the process of starting the vehicle, when the speed is slow (1-3m/s), the time delay will increase suddenly and briefly. Then, in the process of increasing speed, the error will increase linearly with speed and remain at

centimeter-level. The reason for error briefly increase is the peculiarities of the RTK. RTK uses dynamic finite difference technique so that the positioning data will be more precise in the process of the vehicle in motion [31]. The speed interval of 1-3m/s is a process where the vehicle changes its state from stationary to moving. So the error increases for a short time. With the increase of speed, the error decreases and then increases linearly with speed. The max average error is 4.714cm at the speed of 19m/s.

V. CONCLUSION

In this paper, we present a problem with the influence of network time delay on the positioning accuracy of automatic driving; that is, the location of the vehicle has changed when the MEC server receives the positioning message sent by the vehicle. Ignoring this problem will lead to a decrease in positioning accuracy, which will affect the safety and reliability of autonomous vehicles. We propose an algorithm named PCM, which firstly convert GPS coordinates to space rectangular coordinates and estimate one-way time delay, then uses state information and positioning information to predict the real location of the vehicle after a one-way time delay, finally turns the predicted space rectangular coordinates back to GPS coordinates. PCM can correct the positioning error caused by network delay.

The experiment in XuChang and Fangshan shows that PCM has good performance to eliminate the positioning error caused by a one-way time delay. First of all, we test the whole algorithm flow in XuChang automatic driving test field and analyze the distribution characteristics of speed and one-way time delay in the process of driving. We analyze the relationship between speed and one-way time delay and PCM algorithm error. Then we measure the error of PCM in the FangShan driving test field. When we choose the maximum typical time delay (100ms) of 4G and the speed is less than 20m/s, the average error of the PCM algorithm is 2.55cm, and the max error is 4.714cm. So the prediction ability of PCM can fully meet the centimeter-level accuracy required by automatic driving.

REFERENCES

- [1] H. P. Moravec, "The stanford cart and the CMU rover," *Proc. IEEE*, vol. 71, no. 7, pp. 872-884, Oct. 1983.
- [2] M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, *Autonomous Driving*, vol. 10. Berlin, Germany: Springer, 2016, pp. 3-978.
- [3] W. Yijing, L. Zhengxuan, Z. Zhiqiang, and L. Zheng, "Local path planning of autonomous vehicles based on A algorithm with equal-step sampling," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 7828-7833.
- [4] L. Wanninger, "Introduction to network RTK," in *Proc. Int. Assoc. Geodesy Meeting, Work. Group*, vol. 4, no. 1, 2004.
- [5] C.-F. Lin, "Full fusion positioning method for vehicle," U.S. Patent 6240367, May 29, 2001,
- [6] X. Zhu, W. Tian, G. Li, and J. Yu, "Research on localization vehicle based on multiple sensors fusion system," in *Proc. Int. Conf. Comput. Netw., Electron. Autom. (ICCNEA)*, Sep. 2017, pp. 491-494.
- [7] H. Kim and I. Lee, "Localization of a car based on multi-sensor fusion," *ISPRS - Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols.21, pp. 247-250, Sep. 2018.
- [8] F. Caron, E. Duflos, D. Pomorski, and P. Vanheegehe, "GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects," *Inf. Fusion*, vol. 7, no. 2, pp. 221-230, Jun. 2006.

- [9] J. Wang, J. Liu, and N. Kato, "Networking and communications in autonomous driving: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1243–1274, 2nd Quart., 2019.
- [10] V. De Silva, J. Roche, and A. Kondo, "Robust fusion of LiDAR and wide-angle camera data for autonomous mobile robots," 2017, *arXiv:1710.06230*. [Online]. Available: <http://arxiv.org/abs/1710.06230>
- [11] K. Yoneda, N. Hashimoto, R. Yanase, M. Aldibaja, and N. Suganuma, "Vehicle localization using 76GHz omnidirectional millimeter-wave radar for winter automated Driving*," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 971–977.
- [12] U. Montanaro, S. Dixit, S. Fallah, M. Dianati, A. Stevens, D. Oxtoby, and A. Mouzakitis, "Towards connected autonomous driving: Review of use-cases," *Vehicle Syst. Dyn.*, vol. 57, no. 6, pp. 779–814, Jun. 2019.
- [13] S. Ma, F. Wen, X. Zhao, Z.-M. Wang, and D. Yang, "An efficient V2X based vehicle localization using single RSU and single receiver," *IEEE Access*, vol. 7, pp. 46114–46121, 2019.
- [14] F. Wang, D. Guan, L. Zhao, and K. Zheng, "Cooperative V2X for high definition map transmission based on vehicle mobility," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.
- [15] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 2, no. 3, pp. 194–220, Sep. 2017.
- [16] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 74–80, Feb. 2014.
- [17] G. Giordano, M. Segata, F. Blanchini, and R. L. Cigno, "A joint network/control design for cooperative automatic driving," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2017, pp. 167–174.
- [18] D. T. Sandwell, "Reference earth model-WGS84," *Scripps Inst. Oceanogr.*, 2002.
- [19] S. P. Drake, "Converting GPS coordinates [phi, lambda, h] to navigation coordinates (ENU)," Tech. Rep., 2002.
- [20] A. Soloviev, "Tight coupling of GPS and INS for urban navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 46, no. 4, pp. 1731–1746, Oct. 2010.
- [21] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [22] P. H. Dana, "Global Positioning System (GPS) time dissemination for real-time applications," *Real-Time Syst.*, vol. 12, no. 1, pp. 9–40, 1997.
- [23] W. Ding, J. Wang, Y. Li, P. Mumford, and C. Rizos, "Time synchronization error and calibration in integrated GPS/INS systems," *ETRI J.*, vol. 30, no. 1, pp. 59–67, Feb. 2008.
- [24] J. Mannermaa, K. Kalliomiaki, T. Mansten, and S. Turunen, "Timing performance of various GPS receivers," in *Proc. Joint Meeting Eur. Freq. Time Forum IEEE Int. Freq. Control Symp.*, 1999, pp. 287–290.
- [25] R. Pi. (2015). *Raspberry Pi 3 Model B*. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/> {Använd 10 02 2016}
- [26] S. Monk, *Raspberry Pi Cookbook: Software and Hardware Problems and Solutions*. Newton, MA, USA: O'Reilly Media, 2016.
- [27] P. Sarolahti and A. Kuznetsov, "Congestion control in Linux TCP," in *Proc. USENIX Annu. Tech. Conf.*, 2002, pp. 49–62.
- [28] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and solutions," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. S27–S32, Mar. 2005.
- [29] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.
- [30] F. Khan, *LTE for 4G Mobile Broadband: Air Interface Technologies and Performance*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
- [31] C. Mekik and M. Arslanoglu, "Investigation on accuracies of real time kinematic GPS for GIS applications," *Remote Sens.*, vol. 1, no. 1, pp. 22–35, 2009.



YUHAO LING received the bachelor's degree in communication engineering from Beijing Jiaotong University, in 2018. He is currently pursuing the master's degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research fields mainly include data transmission and cooperative positioning in automatic driving.



XINGHE CHU received the bachelor's degree in electronic engineering from Dalian Maritime University, in 2017. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include radio-based simultaneous localization and mapping and cooperative localization and navigation.



ZHAOMING LU received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), in 2012. He is currently an Associate Professor with the School of Information and Communication Engineering, BUPT. His research interests include open wireless networks, QoE management in wireless networks, software defined wireless networks, and cross-layer design for mobile video applications and network-assisted autonomous driving.



LUHAN WANG received the Ph.D. degree from the Beijing University of Posts and Telecommunications (BUPT), in 2017. In 2016, he visited Eurecom as a Visiting Ph.D. Student. In 2017, he joined the School of Information and Communication Engineering, BUPT, as an Assistant Professor. He currently leads a team in BUPT which is getting involved in OpenAirInterface 5G Core Network development. He also works as a Researcher and a Project Manager with the Joint BUPT-Eurecom Open5G Lab. His research interests include network architecture, network function virtualization, and software-defined networks.



XIANGMING WEN received the B.E., M.S., and Ph.D. degrees in electrical engineering from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He is currently the Vice President of BUPT, where he is also a Professor with the Communication Network Center and the Director of the Beijing Key Laboratory of Network System Architecture and Convergence. In the last five years, he has authored more than 100 published articles. His current research interests include broadband mobile communication theory, multimedia communications, and information processing. He is the Principle Investigator of more than 18 projects, including the National Key Project of Hi-Tech Research and Development Program of China (863 Program) and the National Natural Science Foundation of China. He is also the Vice Director of the Organization Committee of the China Telecommunication Association.

...