SCIENCE CHINA
Information Sciences

• Supplementary File •

Queue estimation for isolated signalized intersections in intelligent vehicle-infrastructure cooperation systems

Yunpeng Wang[1], Ge Guo[2,3*] & Wei Yue[4]

[1]*School of Control Science and Engineering,Dalian University of Technology, Dalian* 116026*, China;*
[2]*State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang* 110004*, China;*
[3]*School of Control Engineering, Northeastern University at Qinhuangdao, Qinhuangdao* 066004*, China;*
[4]*School of Marine Electrical Engineering,Dalian Maritime University, Dalian* 116026*, China*

Appendix A Desired Travel Speed Determination

The Desired travel speed sequence needs to be determined before predicting the completed operational data of vehicles based on model (1) in the queue estimation. For autonomous vehicle and human-driven vehicle, the principle of desired travel speed determination is different.

1) For autonomous vehicle, the desired travel speed is determined by the planning layer in the control system. The decision-making with autonomous vehicle is typically divided into hierarchical layers for managing vehicle on different spatial and temporal scales, as shown in Fig. A1. The routing layer makes route guidance in a large road network. The planning layer plays a role in a specific road segment, e.g., the vicinity of signalized intersection, by giving the desired travel speed according to different traffic conditions and objective in fuel economy. Tracking layers then compute proper throttle or braking torques/power to track the desired travel speed.

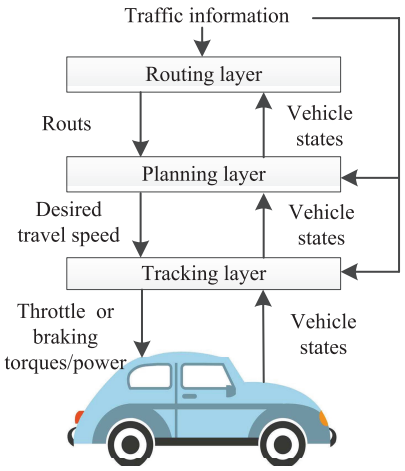

Figure A1 Typical control system for autonomous vehicle.

In the process of passing through the signalized intersection, the sequence of desired travel speed for autonomous vehicle can be directly calculated based on the intelligent maneuvers, using the collected vehicle states and traffic conditions.

* Corresponding author (email: geguo@yeah.net)

2) For human-driven vehicle, desired travel speed is the driver intent in the driving and unable to be directly collected. In this, we use the position of the driver pedal, $p(k)$, to describe the driving intent of human driver. Positive pedal value means the throttle pedal position and negative pedal value means the braking pedal position. Then, the desired travel speed is obtained as

$$v^{itd}(k) = v(k) + \kappa \cdot p(k), \tag{A1}$$

where, κ is a parameter to reflect dynamic characteristics of vehicle.

In the process of approaching the signalized intersection, the intent pedal position is predicted based on the history pedal position, real-time vehicle states, and traffic information. We use a learning-based predict method to infer drivers pedal position by combining a Gaussian mixture model (GMM) with a hidden Markov model (HMM), which is improved from the method in [1]. The GMM is used to model stochastic relationships among variables, while the HMM is applied to infer driver pedal positions based on the GMM. The inputs to the GMM-HMM model include: 1) history and current speed $\{v(1), \cdots, v(k)\}$; 2) history pedal position $\{p(1), \cdots, p(k-1)\}$; 3) history and current space headway $\{\Delta l(1), \cdots, \Delta l(k)\}$; 4) history and current relative distance between vehicle and intersection $\{\Delta d(1), \cdots, \Delta d(k)\}$; 5) history and current signal phase and timing information $\{spat(1), \cdots, spat(k)\}$, where, $spat(k) = 1$ indicates the signal light is green at time k, and $spat(k) = 2$ indicates the signal light is red at time k.

The proposed human driver model is presented generally as

$$\hat{p}(k) = \mathcal{D}\left(\{\chi(1), \cdots, \chi(k-1)\}, x(k)\right). \tag{A2}$$

The equation (A2) is to generate an pedal position according to the current inputs, $[x(k) = [v(k), \Delta l(k), \Delta d(k), spat(k)]^T$, and the history information, $\{\chi(1), \cdots, \chi(k-1)\}$, where $\chi(k) = [x(k), p(k)]^T$.

In practical, the data $\chi(k)$ is considered to obey Gaussian distribution in the GMM,

$$\chi(k) \sim \mathcal{G}_i\left(\mu_i, \sigma_i\right), \tag{A3}$$

where μ_i and σ_i are mean and covariance of the ith Gaussian distribution $\mathcal{G}_i$. The joint probability density function between the current inputs $x(k)$ and the pedal position $p(k)$ is presented in the form of a multivariate Gaussian regression function,

$$\begin{aligned} P(\chi(k) \Big| \{\mu_i, \sigma_i\}_{i=1}^M) &= \sum_{i=1}^M \pi_i \cdot \mathcal{G}_i\left(\chi(k) \,|\mu_i, \sigma_i\right) \\ &= \sum_{i=1}^M \pi_i \cdot \frac{1}{(2\pi)^{d/2}|\sigma_i|^{1/2}} \cdot \exp\left\{-\frac{1}{2}(\chi(k)-\mu_i)^T(\sigma_i)^{-1}\left(\chi(k)-\mu_i\right)\right\}, \end{aligned} \tag{A4}$$

where, π_i is the prior probability and $\sum_{i=1}^M \pi_i = 1$. $M \in \mathbb{N}^+$ is the number of Gaussian components.

Based on a data set collected form a particular driver, the parameters μ_i, σ_i, and π_i can be estimated by expectation maximization (EM) method. The data set is a d-dimension sequence, $\chi = \{\chi(k)\}, k = 1, \cdots, N$, with $\chi(k) \in \mathbb{N}^{d\times 1}$. N is the length of training data set. Assuming μ_i^s, σ_i^s, and π_i^s are the estimation at iteration step s. The k-means clustering algorithm is used to obtain the initial value μ_i^0, σ_i^0, and π_i^0 at iteration step $s = 0$. Then, the optimal parameter μ_i^*, σ_i^*, and π_i^* can be estimated until the log-likelihood function is convergent or reaches the maximum steps, which is formulated as

$$(\mu_i^*, \sigma_i^*, \pi_i^*) = \arg \max_{\mu_i, \sigma_i, \pi_i} \mathcal{L}\left(\mu_i, \sigma_i, \pi_i\right) = \arg \max_{\mu_i, \sigma_i, \pi_i} \sum_{k=1}^N \log\left(\chi(k) \,|\mu_i, \sigma_i, \pi_i\right). \tag{A5}$$

After learning the GMM parameters, we use the HMM approach to generate the pedal position sequence based on the learned GMM driver model. The joint distribution $P(x(k), p(k) \,|\mu_i, \sigma_i, \pi_i)$ is used to generate the pedal position in a continuous HMM of M modes. $x(k)$ is the observable state, and $p(k)$ is the unobservable state we need to predict. Here, each Gaussian of GMM is considered as a mode of HMM. $\mu = \{\mu_i\}_{i=1}^M$ and $\sigma = \{\sigma_i\}_{i=1}^M$ are the mean and covariance matrix of the Gaussian distribution of the HMM.

The pedal position at time k can be predicted, given the history information, $\{\chi(1), \cdots, \chi(k-1)\}$ and the observed state $x(k)$ at time k, by using

$$\begin{aligned} \hat{p}(k) &= E\left(p(k) \,|\{\chi(1), \cdots, \chi(k)\}\right) \\ &= \sum_{i=1}^M \psi_i(x(k)) \left[\mu_i^p + \sigma_i^{p,x}(\sigma_i^{p,p})^{-1}(x(k) - \mu_i^x)\right], \end{aligned} \tag{A6}$$

where

$$\mu_i = \left[\mu_i^x, \mu_i^p\right]^T, \sigma_i = \begin{bmatrix} \sigma_i^{x,x} & \sigma_i^{x,p} \\ \sigma_i^{p,x} & \sigma_i^{p,p} \end{bmatrix}^T, \tag{A7}$$

and $\psi_i(x(k))$ is the HMM forward variable, computed as the probability of being in state i at time k, given by,

$$\psi_i(x(k)) = \frac{\left(\sum_{j=1}^M \psi_j(x(k-1)) \cdot \varphi_{j,i}\right) \cdot \mathcal{G}_i\left(x(k) \left|\mu_i^x, \sigma_i^{x,x}\right.\right)}{\sum_{l=1}^M \left(\sum_{j=1}^M \psi_j(x(k-1)) \cdot \varphi_{j,l}\right) \cdot \mathcal{G}_l\left(x(k) \left|\mu_l^x, \sigma_l^{x,x}\right.\right)}. \tag{A8}$$

Here, the initial value at time $k = 1$ is calculated by

$$\psi_i(x(1)) = \frac{\pi_i \cdot \mathcal{G}_i\left(x(1) \left|\mu_i^x, \sigma_i^{x,x}\right.\right)}{\sum_{j=1}^M \pi_i \cdot \mathcal{G}_i\left(x(1) \left|\mu_i^x, \sigma_i^x\right.\right)}. \tag{A9}$$

Appendix B Training of Traffic Flow Model

The car-following function $f(\cdot)$ and car-following parameters θ in the proposed traffic flow model (1) are formulated as follows,

$$\begin{aligned} &f\left(v(k),\Delta l(k),h^{itd}(k),v^{itd}(k)\,|\theta\right)=\phi_1\cdot f_1\left(v(k),\left(h^{itd}(k)-\Delta l(k)\right)|\theta\right)+\phi_2\cdot f_1\left(v^{itd}(k)-v(k)\,|\theta\right)\\ &=\phi_1\cdot\left\{\frac{a_{\min}}{1+e^{-k_1\cdot\left[\Delta l(k|\theta)-h^{itd}(k|\theta)\right]-k_2}}\right\}+\phi_2\cdot\left\{\frac{k_5\cdot a_{\min}}{1+e^{-k_3\cdot\left[v(k|\theta)-v^{itd}(k|\theta)\right]-k_4}}+\frac{k_8\cdot a_{\max}}{1+e^{-k_6\cdot\left[v(k|\theta)-v^{itd}(k|\theta)\right]-k_7}}\right\}+k_9, \end{aligned} \tag{B1a}$$

$$\theta=\left\{\begin{array}{ccccccccccccc}\phi_1 & \phi_2 & \alpha & \beta & \delta & k_1 & k_2 & k_3 & k_4 & k_5 & k_6 & k_7 & k_8 & k_9\end{array}\right\}, \tag{B1b}$$

where ϕ_1 and ϕ_2 are the weights of safety and intent, respectively; $a_{\min}$ and $a_{\max}$ are the limits of acceleration; $k_1,\cdots,k_8$ are coefficients of the sigmoid functions; k_9 is the bias term. $f_1(\cdot)$ describes the relationship between the acceleration and space headway. If the current space headway is approaching to the desired safe space headway, the speed of the vehicle should be decreased to ensure that the space headway will not be smaller than the desired safe space headway of vehicle. While, $f_2(\cdot)$ describes that acceleration and deceleration of vehicle is related to the difference between its current speed and desired travel speed. If the current speed is lower than desired travel speed, vehicle should accelerate to ensure the speed will approach the desired travel speed, otherwise, vehicle should decelerate. Generally, the vehicle will track the desired travel speed on the premise of ensuring the desired safe space headway in the approaching to the intersection.

We use a three-layer feedforword neural network (FNN) to learn the function $f(\cdot)$ and get the parameters θ. The mathematical formulation of a three-layer FNN is

$$z_o=S_o\left(\sum_{h=1}^{m}\left[b_{o,h}\cdot S_h\left(\sum_{e=1}^{n}\left(a_{h,e}\cdot x_e\right)+bias_h\right)\right]+bias_o\right),o=1,\cdots,p, \tag{B2}$$

where z_o is the oth output variable; x_i is the ith input variable; $S_o(\cdot)$ and $S_h(\cdot)$ are the transfer function; $S_o(\cdot)$ is chosen as linear function and $S_h(\cdot)$ is chosen as sigmoid function; $a_{h,i}$ and $b_{o,h}$ are the matrices of weights of the input-hidden and the hidden-output layers, respectively; and *bias* denotes the bias term of the FNN.

Due to the difference principle of the desired travel speed for human driver and autonomous control system (details are described in Appendix A), the car-following model of the human-driven vehicle and the autonomous vehicle will be trained differently.

1) For autonomous vehicle, the input variables include: current space headway, speed, square of speed and desired travel speed. The output variable is acceleration. The numbers of neurons of the input layer, hidden layer and output layer are 4, 3 and 1, respectively (i.e., $n=4$, $m=3$, $p=1$). Based on the deformation of formula (A1), we can get the weight matrixes for autonomous vehicles model training as follows,

$$\mathbf{a}=\begin{bmatrix}a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4}\\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4}\\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4}\end{bmatrix}=\begin{bmatrix}k_1 & -k_1\alpha & -k_1\beta & 0\\ 0 & k_3 & 0 & -k_3\\ 0 & k_6 & 0 & -k_6\end{bmatrix},$$

$$\mathbf{Bias}_h=\begin{bmatrix}-k_1\delta+k_2 & k_4 & k_7\end{bmatrix}^T,$$

$$\mathbf{Bias}_o=[k_9],$$

$$\mathbf{b}=\begin{bmatrix}b_{1,1} & b_{1,2} & b_{1,3}\end{bmatrix}=\begin{bmatrix}\phi_1 a_{\min} & \phi_2 k_5 a_{\min} & \phi_2 k_8 a_{\max}\end{bmatrix}.$$

2) For human-driven vehicle, we use the position of the driver pedal $p_i(k)$ as the input variable, instead of desired travel speed. The weight matrixes for human-driven vehicles model training can be got as follows,

$$\mathbf{a}=\begin{bmatrix}a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4}\\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4}\\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4}\end{bmatrix}=\begin{bmatrix}k_1 & -k_1\alpha & -k_1\beta & 0\\ 0 & 0 & 0 & -k_3\kappa\\ 0 & 0 & 0 & -k_6\kappa\end{bmatrix},$$

$$\mathbf{Bias}_h=\begin{bmatrix}-k_1\delta+k_2 & k_4 & k_7\end{bmatrix}^T,$$

$$\mathbf{Bias}_o=[k_9],$$

$$\mathbf{b}=\begin{bmatrix}b_{1,1} & b_{1,2} & b_{1,3}\end{bmatrix}=\begin{bmatrix}\phi_1 a_{\min} & \phi_2 k_5 a_{\min} & \phi_2 k_8 a_{\max}\end{bmatrix}.$$

The back-propagation algorithm is used for model training to obtain the optimal θ. The principle of back-propagation algorithm is reducing the total error through modifying the gradient weights and biases, which is formulated as,

$$E=\frac{1}{D_M}\sum_{Di=1}^{D_M}\sum_{o=1}^{p}(r_o(Di)-z_o(Di))^2, \tag{B3}$$

where E is the total error; $r_o(Di)$ is the actual output; and D_M is the sample number in the set of training data.

The modification of weights and bias terms in the FNN can be respectively formulated as

$$\Delta a_{h,e}=-\eta\frac{\partial E}{\partial a_{h,e}},\Delta bias_h=-\eta\frac{\partial E}{\partial bias_h},e=1,\cdots,n,h=1,\cdots,m, \tag{B4a}$$

$$\Delta b_{o,h} = -\eta \frac{\partial E}{\partial b_{o,h}}, \Delta bias_o = -\eta \frac{\partial E}{\partial bias_o}, h = 1, \cdots, m, o = 1, \cdots, p, \tag{B4b}$$

where η denotes the learning rate.

The error signal terms for the neurons in the hidden and output layers are defined respectively as,

$$\omega_h = -\frac{\partial E}{\partial u_h}, \omega_o = -\frac{\partial E}{\partial u_o}, h = 1, \cdots, m, o = 1, \cdots, p, \tag{B5}$$

where,

$$u_h = \sum_{e=1}^{n} \left(a_{h,e} \cdot x_e\right) + bias_h, u_o = \sum_{h=1}^{m} \left[b_{o,h} \cdot S_h \left(\sum_{e=1}^{n} \left(a_{h,e} \cdot x_e\right) + bias_h\right)\right] + bias_o. \tag{B6}$$

Then, the gradient of the total error to weights and bias terms in the hidden and output layers is developed respectively as,

$$\frac{\partial E}{\partial a_{h,e}} = \frac{\partial E}{\partial u_h} \cdot \frac{\partial u_h}{\partial a_{h,e}} = -\omega_h x_e, \frac{\partial E}{\partial bias_h} = \frac{\partial E}{\partial u_h} \cdot \frac{\partial u_h}{\partial bias_h} = -\omega_h, e = 1, \cdots, n, h = 1, \cdots, m, \tag{B7a}$$

$$\frac{\partial E}{\partial b_{o,h}} = \frac{\partial E}{\partial u_o} \cdot \frac{\partial u_o}{\partial b_{o,h}} = -\omega_o S_h\left(u_h\right), \frac{\partial E}{\partial bias_o} = \frac{\partial E}{\partial u_o} \cdot \frac{\partial u_o}{\partial bias_o} = -\omega_o, h = 1, \cdots, m, o = 1, \cdots, p, \tag{B7b}$$

Furthermore, Eq. (B5) can be obtained as,

$$\omega_o = -\frac{\partial E}{\partial u_o} = -\frac{\partial E}{\partial z_o} \cdot \frac{\partial z_o}{\partial u_o} = \frac{2}{D_M} \sum_{ta=1}^{D_M} \sum_{o=1}^{p} \left(r_o(k_{ta}) - z_o(k_{ta})\right) S'_o(u_o), o = 1, \cdots, p, \tag{B8a}$$

$$\omega_h = -\frac{\partial E}{\partial u_h} - \frac{\partial E}{\partial u_o} \cdot \frac{\partial u_o}{\partial u_h} = \omega_o b_{o,h} S'_h(u_h), h = 1, \cdots, m. \tag{B8b}$$

Finally, the weights and bias terms can be modified at each step as,

$$a_{h,e}(k+1) = a_{h,e}(k) + \eta \omega_h x_e(k), bias_h(k+1) = bias_h(k) + \eta \omega_h, e = 1, \cdots, n, h = 1, \cdots, m, \tag{B9a}$$

$$b_{o,h}(k+1) = b_{o,h}(k) + \eta \omega_o S_h\left(u_h\right), bias_o(k+1) = bias_o(k) + \eta \omega_o, o = 1, \cdots, p, h = 1, \cdots, m. \tag{B9b}$$

Appendix C Predicting the Number of Vehicles Arriving and Leaving the Queue

Based on the analysis of queue forming and discharging, the number of vehicles arriving and leaving the queue in the sampling interval $[k, k+1]$ can be predicted as

$$N_{in}(k+1\,|k\,) = F_{in}\left(v_0^{itd}(k), \left\{v_i^{itd}(k), i \in \mathrm{A}_k\right\}\right), \tag{C1a}$$

$$N_{out}(k+1\,|k\,) = F_{out}\left(v_0^{itd}(k), \left\{v_i^{itd}(k), i \in \mathrm{H}_k\right\}\right), \tag{C1b}$$

where A_k is the index set of autonomous vehicle approaching the queue at time k, H_k is the index set of autonomous vehicle stopping in the queue and left from the queue at time k, $F_{in}(\cdot)$ and $F_{out}(\cdot)$ are the mapping functions, which are described as Algorithms C1 and C2 as follows, respectively.

Algorithm C1 Pseudocode for $F_{in}(\cdot)$

Input: Current time k; The desired travel speed of virtual vehicle and autonomous vehicles at time k, $v_0^{itd}(k)$,$v_p^{itd}(k)$,$p \in A_k$;

Output: The number of vehicles which arrive at the tail of queue from time k to $k+1$,$N_{in}(k+1\,|k\,)$;

$n \Leftarrow 0$;$h \Leftarrow 1$
while $h \leqslant M$ **do**
  if $v_h(k) == 0$ **then**
    $n \Leftarrow h$;
  end if
  $h \Leftarrow h+1$;
end while
if $n \in [1, M]$ **then**
  $h \Leftarrow 1$;
  while $h \leqslant M-n$ **do**
    Calculate $l_{n+h}^{DP}(k)$, $l_{n+h}(k+1)$ based on equation (1-3)
    if $l_{n+h}(k+1) < l_{n+h}^{DP}(k)$ **then**
      $Num_{in}(k+1\,|k\,) \Leftarrow h-1$
      break
    end if
    $h \Leftarrow h-1$;
  end while
else
  $N_{in}(k+1\,|k\,) \Leftarrow 0$
end if

Algorithm C2 Pseudocode for $F_{out}(\cdot)$

Input: Current time k; The desired travel speed of virtual vehicle and autonomous vehicles at time k, $v_0^{itd}(k)$,$v_p^{itd}(k)$,$p \in A_k$;

Output: The number of vehicles which arrive at the tail of queue from time k to $k+1$,$N_{in}(k+1\,|k\,)$;

$m \Leftarrow 0$;$h \Leftarrow M$
while $h \geqslant 1$ **do**
  if $l_h(k) \leqslant l_{stop}$ **then**
    $m \Leftarrow h$;
  end if
  $h \Leftarrow h-1$;
end while
if $m \in [1, M]$ **then**
  $h \Leftarrow 1$;
  while $h \leqslant n-m$ **do**
    Calculate $l_{m+h}(k+1)$ based on equation (1)
    if $l_{m+h}(k+1) < l_{stop}$ **then**
      $Num_{in}(k+1\,|k\,) \Leftarrow h-1$
      break
    end if
    $h \Leftarrow h+1$;
  end while
else
  $N_{out}(k+1\,|k\,) \Leftarrow 0$
end if

Appendix D Experiment Simulations

The models and algorithms proposed in this paper are tested in microscopic traffic simulation with cooperative vehicle intersection control (CVIC) environment. The simulation scenario considered in this paper is a single lane road with an isolated signalized intersection at location 0. Overtaking, lane-changing, and lateral motions of vehicles are not considered. Meanwhile, pedestrian are also not considered.

The longitudinal motion of human-driven vehicle is simulated by MATLAB/Simulink interfaced with TORCS [2] to provide a dynamic model. All the human driver data used in this paper are collected in the TORCS. In the learning process, 80% input samples are used for training and 20% input samples for model validation. For GMM-HMM, the number of Gaussian components M is set as 20, and length of training data N is set as 30. For FNN, the maximum

number of training iterations, the minimum performance gradient, and the learning rate η are set to 1000, 10×10^{-3}, and 0.01, respectively. The longitudinal motion of autonomous vehicle is simulated in MATLAB/Simulink as ref. [3], with an asynchronous service framework to achieve the cooperative simulating with human-driven vehicle in the TORCS. To get CVIC environment, the adaptive control strategy of traffic signal in the simulation is chosen as ref. [4], and intelligent maneuver of autonomous vehicles is chosen as ref. [5]. Maximum acceleration and deceleration are assumed to be $a_{\max} = 2m/s^2$ and $a_{\min} = -3m/s^2$, respectively. The parameters set in the traffic flow model used for prediction are $\theta = \{0.91, 0.79, 1.21, 0.20, 13.09, -1.52, 2.92, -1.48, -4.73, 0.81, 1.54, -3.88, 0.72, -0.09\}$.

We compare the proposed model in this paper with the queue length estimation model in [6]. To compare the performance of the two models, we use the real-time queue location (head and tail) and the Mean Absolute Error Rate (MAER). Real-time queue location trajectory can describe the estimation results intuitively with comparing to the observed vehicle trajectory. For proposed model, the queue location is transformed from the vehicle locations calculation in Algorithms C1 and C2. While, for model in [6], the real-time queue location can be got from the queue forming wave and discharging wave. MAER of estimated queue length to observations in N experiments is calculated as follows

$$MAER = \frac{1}{N} \sum\nolimits_{p=1}^{N} \left| \frac{\bar{x}_p - x_p^*}{x_p^*} \right|, \tag{D1}$$

where, $\bar{x}_p$ is the estimated value and x_p^* is the observed value at pth experiment.

In order to more clearly compare with the existing methods, all simulation cases are multi-step estimation in a complete signal cycle. The sampling interval is chosen as 1s. The estimated queue location and length in the signal cycle is multistep calculated based on the individual data at k_0. In the plotting, take one out of every five to make the figure clearer and easily readable. The computational time of multistep estimation for a signal cycle is less than 2s using a PC (with Intel(R) Core i7C4790 CPU @3.6GHZ).

A. Under-saturated case

In this case, the proposed model is tested comparing with model in [6] in under-saturated traffic scenario. The traffic flow rate and density are set as 540 veh/h and 11 veh/km, respectively. The penetration rate of autonomous vehicle in the traffic flow is 30%. We conduct two experiments with the same initial information, i.e., same original phase and time information of traffic signal, and the same initial location and speed of vehicle. In one cycle, the signal original phase and time without adaptive strategy: the signal turns to red as the beginning of cycle at time 30s; turns to green at time 65s; turns back to red as the ending of cycle at time 100s. The initial location and speed of vehicle are randomly generated according to the selected traffic flow rate.

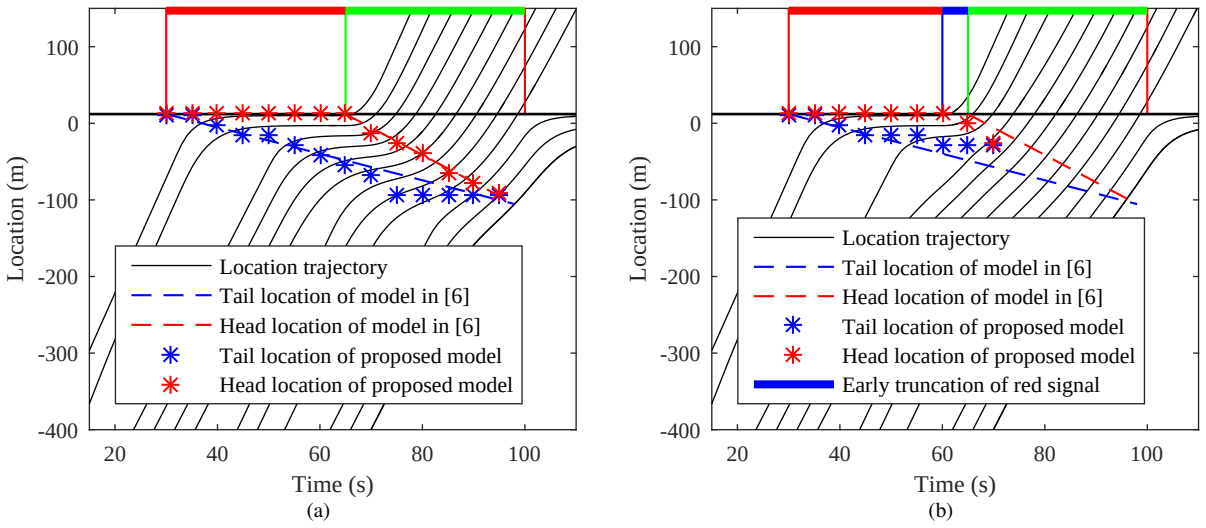

Figure D1 Estimation results comparison of two models in under-saturated case. (a) Conventional experiment; (b) CVIC experiment.

The estimated queue locations of two model and observed vehicles trajectory in two experiments are shown in Fig. D1. Fig. D1(a) describes the estimation results of two models in the conventional experiment: signal duration is fixed with initial set and there are no autonomous vehicles in the traffic flow. The solid lines are the actual vehicle trajectory. The asterisks and dotted lines are the queue locations estimated by proposed model and model in [6], respectively. The head location and tail location of queue have intersected at some time of green signal. This means the queue formed at the red signal has completely dissipated before next signal cycle. The estimated queue locations of two models in the conventional experiment are match the observation very well, and the error of estimation are all small. Fig. D1(b) shows the estimation results of two models in the CVIC experiment: the red signal has an early truncation (5s) based on the adaptive strategy

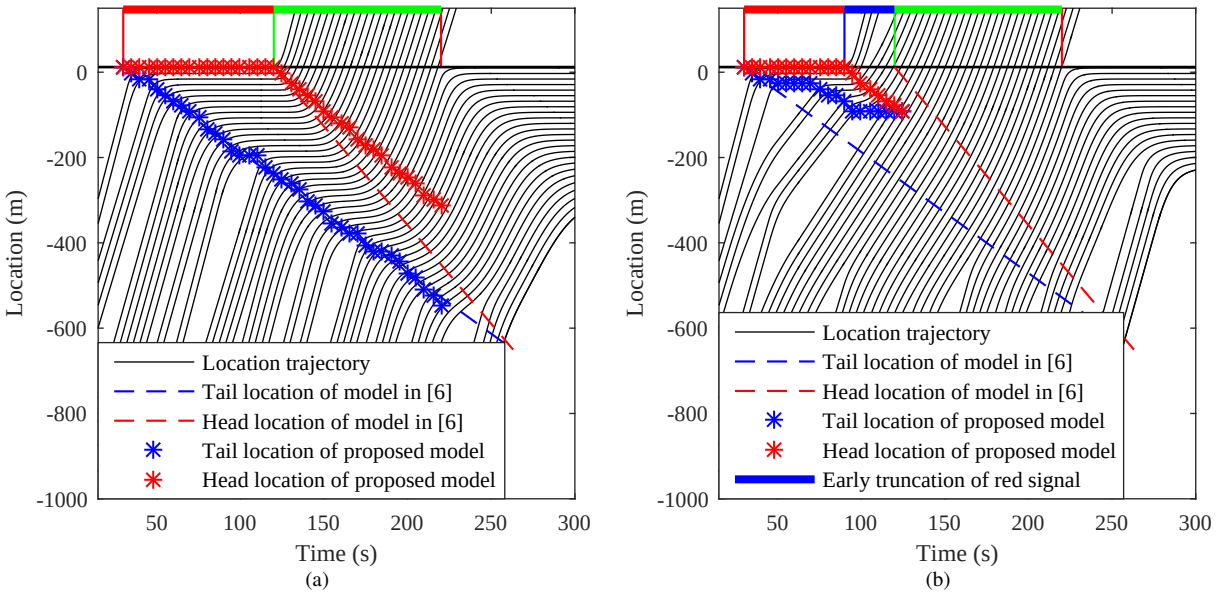

Figure D2 Estimation results comparison of two models in over-saturated case. (a) Conventional experiment; (b) CVIC experiment.

and the autonomous vehicles apply deceleration maneuvers to avoid stopping at the intersection. The blue solid line is the new switching time of signal from red to green based on the adaptive traffic signal control. In the CVIC experiment, the estimated result of model in [6] has a large error. However, the performance of proposed model is well, and the error of estimation is still small.

Moreover, in our model, a queued vehicle is defined using whether the arrival of decision point (DP) will happen in the approaching to the tail of queue. The arrival of DP does not mean that the vehicle is bound to stop. It is also possible that the vehicle has a sharp deceleration and then accelerate back. From the estimation results of proposed model in Fig. D1(a), we can find that clearly: the proposed model estimates not only the number of completely stopped vehicles, but also the vehicles which have sharp deceleration-and-acceleration movement. While, in Fig. D1(b), since the autonomous vehicle has adopted the intelligent maneuver, i.e., vehicle 4, its trajectory is smooth and there is no sharp deceleration-and-acceleration movement. So, vehicle 4 is not estimated as queued vehicle by the proposed model, as shown as the tail location of proposed model.

The location trajectory of vehicle in the CVIC experiment is very different from the conventional experiment, since the control inputs of traffic signal and autonomous vehicles have impacts on the driving of vehicles. The model in [6] is unable to considered influence of the control inputs in the estimation and can only be applied in the estimation with uniform arrivals of traffic flow which is considered as the average arrival rate of the cycle. While, the proposed model in this paper has good adaptability in different traffic flows. The influence of control inputs of traffic signal and autonomous vehicles on queuing at intersection can be well reflected by the proposed model.

B. Over-Saturated case

In this case, the proposed model is tested comparing with model in [6] in over-saturated traffic scenario. The traffic flow rate and density are set as 800 veh/h and 16 veh/km, respectively. The penetration rate of autonomous vehicle in the traffic flow is 50%. In the same way, we conduct two experiments in conventional and CVIC environments separately. In one cycle, the signal original phase and time without adaptive strategy: the signal turns to red as the beginning of cycle at time 30s; turns to green at time 120s; turns back to red as the ending of cycle at time 220s. The estimated queue locations of two model and observed vehicles trajectory in two experiments are shown in Fig. D2. In the conventional experiment, the head location trajectory and tail location trajectory are not intersected before the beginning of next signal cycle. This means the queue that is formed during the red indication cannot be completely served in the same cycle. The model in [6] has well estimation results only in the conventional experiment and the proposed model has good adaptability in both experiments.

C. Discussions

We discuss in this subsection the impacts of different control input on the estimation of models. First, we ran the estimation model repeatedly with different penetration rate of autonomous vehicle and original phase and time of traffic signal. The relationship between MAER of estimation and penetration rate are described in Fig. D3. When the penetration rate is low, the intelligent maneuver of autonomous vehicles is difficult to affect the accuracy of the model in [6] in mean sense. As the penetration rate rises, the impact will become larger and reach a maximum. This means that: when the penetration rate of autonomous vehicle in traffic flow reaches a certain degree, the regulation and improvement of intelligent

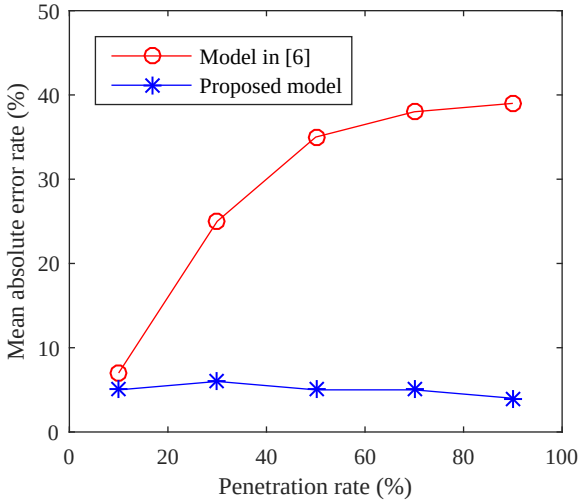

Figure D3 MAER vs. penetration rate of autonomous vehicle.

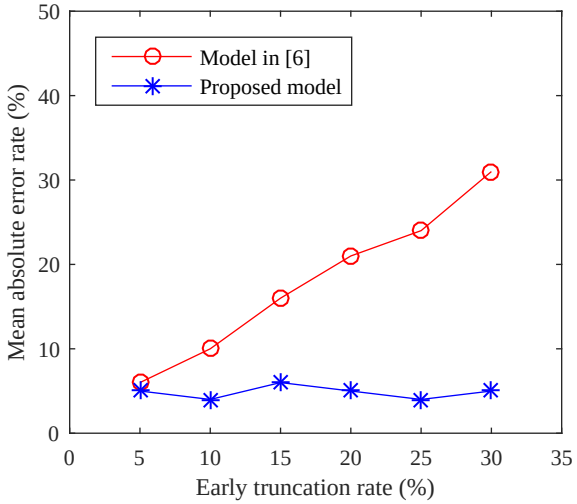

Figure D4 MAER vs. early truncation rate of red signal.

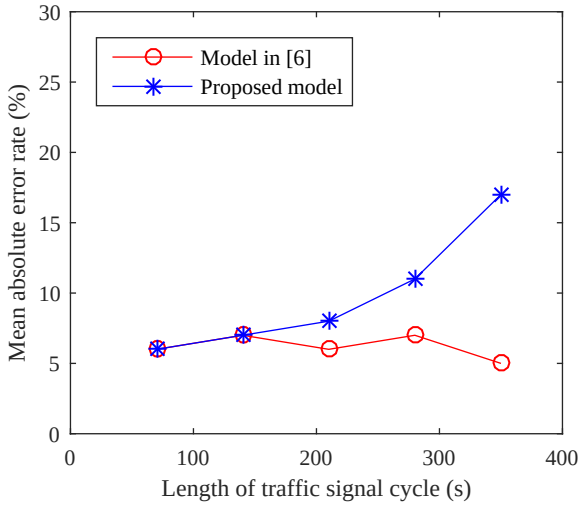

Figure D5 MAER vs. length of traffic signal cycle.

maneuver to traffic flow is close to the maximum. The relationship between MAER of estimation and early truncation rate are also identified through plentiful repeated experiment and statistical analysis, which are depicted in Fig. D4. Early truncation of red signal refers to cutting the duration of red signal in order to provide green signal more time. The early truncation rate (ETR) is calculated as follows,

$$ETR = \frac{\Delta t_r}{T_r}, \tag{D2}$$

where, Δt_r is the early truncation, T_r is the original red signal duration. In practice, the ETR has a maximum limit. From Fig. D4, we can get that the control input of adaptive traffic signal can linearly affect the estimation accuracy. If the estimation model is designed with pre-set traffic signal, it cannot be applied in the traffic scenario with CVIC. The longer the actual queue length is, the larger the control inputs of the signal is, and the greater the estimation error is.

As Fig. D3 and D4 shown, the MAER of proposed model in this paper is not affected by the control inputs of traffic signal and autonomous vehicles. However, when the signal cycle is long, the multistep estimation of queue length for whole cycle will have a large error relatively as shown in Fig. D5. This is because the accumulation of calculation error in multi-step estimation. When the length of signal cycle is not very long, the multistep estimation can be applied as online prediction for the adaptive control.

References

1 Wang W, Zhao D, Xi J, et al. Development and evaluation of two learning-based personalized driver models for car-following behaviors. In: American Control Conference, Seattle, USA, 2017. 1133-1138

2 Onieva E. Pelta D A, Alonso J, et al. A modular parametric architecture for the TORCS racing engine. In: IEEE Symposium on Computational Intelligence and Games, Milano, Italy, 2009. 256-262

3 Asadi B, Vahidi A. Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time. IEEE Trans Control Syst Technol, 2011, 19: 707-714

4 Cai C, Wang Y, Geers G. Vehicle-to-infrastructure communication-based adaptive traffic signal control. IET Intell Transp Syst, 2013, 7: 351-360

5 Yang H, Rakha H, Ala M V. Eco-Cooperative Adaptive Cruise Control at Signalized Intersections Considering Queue Effects. IEEE Trans Intell Transp Syst, 2016, 18: 1575-1585

6 Ban X, Hao P, Sun Z. Real time queue length estimation for signalized intersections using travel times from mobile sensors. Transp Res Pt C-Emerg Technol, 2011, 19: 1133-1156