Estimating Traffic Signal Settings and Queue Lengths Using Connected and Autonomous Vehicles Data

Alireza Soltani*, Mohsen Ramezani

School of Civil Engineering, The University of Sydney, Australia
alireza.soltani@sydney.edu.au, mohsen.ramezani@sydney.edu.au
* Corresponding author

Extended abstract submitted for presentation at the Conference in Emerging Technologies in Transportation Systems (TRC-30)
September 02-03, 2024, Crete, Greece

April 30, 2024

Keywords: Connected and Autonomous Vehicles (CAVs), Traffic state estimation, Queue length estimation

1 INTRODUCTION

The increasing prevalence of vehicles equipped with onboard sensors, ranging from conventional vehicles with advanced sensor capabilities to autonomous vehicles (AVs), offers significant potential for traffic state estimation (Li *et al.* (2020)). AVs provide substantial data from their surroundings Chen *et al.* (2022), but their network penetration rate is expected to remain low in the near future. This paper introduces a lane-based queue length estimation method for city streets based on AV data, as existing works in the literature often overlooked multi-lane streets Ramezani & Geroliminis (2015), Guo *et al.* (2019).

In this study, AV data refers to the information autonomous vehicles collect as they move through the network. This includes detailed data about the AV's own status at each time step, as well as information about surrounding vehicles within a specific detection radius. Key data inputs include vehicles' position, speed, and ID.

AVs can detect a specific vehicle at multiple locations and offer a continuous and real-time view of traffic conditions by capturing real-time data of surrounding vehicles. AVs close to intersections, especially those in the queues, can gather data of vehicles entering the intersection from all lanes. The data can be used to reconstruct missing data in vehicle trajectories. Aggregating data from multiple AVs across the network provides opportunities to estimate traffic states even with a low percentage of AVs in the network and enables a comprehensive understanding of queueing dynamics at intersections.

The proposed approach assumes no explicit information regarding signal timings or vehicle arrival distributions. To account for the ultra-low penetration rate of AVs, we developed methods suitable for varying AV penetration rates of 1%, 2%, and 5% in the network. We propose a method for estimating traffic signal states, reconstructing vehicle trajectories and estimating queue length within urban networks. Since an existing dataset with a percentage of AVs in the network is unavailable, we simulated data using AIMSUN software, modelling a network with 16 intersections (4 actuated and 12 fixed-time signal intersections).

2 METHODOLOGY

The proposed methodology begins with a multi-level system for estimating traffic signal state data. Initially, we concentrate on data from one lane at a time, focusing on vehicles approaching an intersection. By identifying AVs on a lane within the visible range of the traffic signal, we can determine the state of the signal for instances where the AV can observe the traffic light. To accommodate low AV penetration rates, we incorporate other data sources, as described in the following paragraphs, to infer traffic states over time.

We use stationary vehicles to discern the red signal state. Let us assume that the speed of the queue discharge shockwave w is derived as in eq. 1, where q_{max} is maximum flow, v_{ff} is free flow speed, and k_{jam} is jam density. We estimate the projected time for each stopped vehicle relative to the intersection by eq. 2 where t_{p} is projected time at intersection, t_{i} is the current time of stopped vehicle, and S_i and S_{int} are the current position of the stopped vehicle and the position of the intersection, respectively. This process allows us to deduce the traffic signal state at time (t_{p}).

$$w = \frac{q_{\mathrm{max}}}{\frac{q_{\mathrm{max}}}{v_{\mathrm{ff}}} - k_{\mathrm{jam}}} \tag{1}$$

$$t_{\mathrm{p}} = t_{\mathrm{i}} + \frac{|S_i - S_{\mathrm{int}}|}{w} \tag{2}$$

Further, we identify green phases by considering various vehicle characteristics associated with green signal states. Firstly, vehicles within their safe stopping distance from the intersection (eq. 3 where d_{s} is safe stopping distance, v_i is the current velocity of the vehicle, t_{r} is the reaction time of the driver, and a_{max} is the maximum deceleration of the vehicle) imply a green signal state by eq. 4, where d_i represents the distance of the vehicle to the intersection. In addition, vehicles recently crossing the intersection and still close to the crossing line indicate additional green signal states.

$$d_{\mathrm{s}} = v_i t_{\mathrm{r}} + \frac{v_i^2}{2a_{\mathrm{max}}} \tag{3}$$

$$\begin{cases} \text{green} & \text{if } d_{\mathrm{s}} < d_i \\ \text{unknown} & \text{otherwise} \end{cases} \tag{4}$$

Having estimated the majority of traffic signal states over time, the next objective is to estimate the remaining unknown signal states. For each lane, we identify conflicting movements at the intersection and estimate their signal states using the same methodology. We infer the subject lane to be red if any conflicting movements are deemed green. Subsequently, we address the unknown times of traffic signals by assigning the closest known traffic state in the time series. Furthermore, we ensure that each green phase lasts for at least 10 seconds and each red phase for at least 30 seconds, though these durations may vary based on intersection configurations.

The second major step of the method is trajectory reconstruction. We start by locating the earliest record in time to reconstruct trajectories for each vehicle's data. If the data represents a moving vehicle, we project it back to when it entered the lane using free-flow speed. For stopped vehicles, we check if there is a vehicle behind them with zero speed. If not, we project the stopped vehicle's data similarly to the previous case. If a stopped vehicle is behind them, we generate a stopping data point for that vehicle's most recent time step. We continue this process until there are no stopped vehicles behind the vehicle.

Next, we generate data for missed time steps by sorting vehicles based on projected entry time. We maintain data continuity by interpolating between existing records or extrapolating based on the last known velocity and acceleration, ensuring that generated speeds do not exceed the maximum permissible speed. We also consider signal states and safe distance to the front vehicle, preventing vehicles from crossing a red signal or having rear-end collisions.

After reconstructing all trajectories, we proceed to estimate queue length. In the queue estimation step, we determine join and leave points for vehicles. Join points signify where vehicles stop, while leave points denote where they start moving again. Partitioning data into cycles, we align trajectories with the intersection position and the end of the red phase, allowing us to regress the head of the queue line. By regression using the start of red and join points, we derive the back of the queue line, from which we identify the queue clearance point. Iteratively refining this estimation, we establish a function representing the back of the queue. The difference between the head and back of queue functions yields the queue length. Finally, we apply the queue length estimation procedure to ground truth data, determining the queue length as a function of time.

3 Results

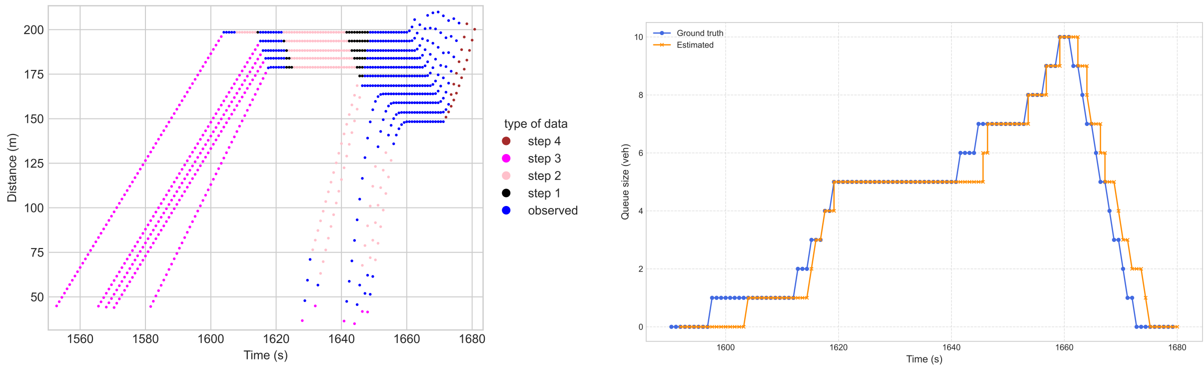
Each signalized intersection featured four directions and 12 lanes. We exported vehicle data from the simulation, including position, speed, acceleration, and ID. We then designated a percentage of vehicles as autonomous and filtered their data along with data from vehicles within their 30-meter detection radius at each timestep. The estimation accuracy of traffic signal settings varies depending on the penetration rate of autonomous vehicles (AVs) in the network and the type of traffic signal system (see Table 1).

Table 1 – *Accuracy of Estimated Signals*

AV Penetration Rate (%)	Actuated Signals	Fixed-Time Signals
5	75%	90%
2	62%	78%
1	50%	65%

Once we have estimated the signal settings, we reconstruct trajectories (see figure 1a). We refer to the existing data in our dataset as observed data. The reconstruction process involves multiple levels for each vehicle's data. In step 1, we address data points where vehicles are stopped (i.e., zero speed). If there is a stopped front vehicle, we generate additional stopped data for the next time step. Similarly, if a vehicle was stopped in a previous time step, we generate another stopped data point for that timestep. In step 2, we handle vehicles with gaps in their data by interpolating the missing data between the boundary data points. Step 3 involves locating the earliest record in terms of time and reconstructing the vehicle's trajectory back to when it entered the lane using free-flow speed. In step 4, we consider the last data for each vehicle. If the vehicle has not crossed the crossline on that point, we generate data for the next step based on the velocity, acceleration, and traffic signal state.

After reconstructing the data, we estimate the queue size. An example of the comparison between the ground truth queue size and the estimated queue size is illustrated in figure 1b. To measure the accuracy of the queue length estimation, we use the mean absolute error (MAE) with the unit of vehicles. Table 2 shows the MAE across all cycles.



(a)

(b)

Figure 1 – *The sample for one cycle with 5% AV in the network (a) observed data and reconstructed data, (b) ground truth and estimated queue size with MAE=0.34 (veh).*

Table 2 – *Mean Absolute Error (MAE) of Queue Length Estimation*

AV Penetration rate (%)	Actuated signals	Fixed-time signals
5	1.25	0.67
2	1.48	0.83
1	2.15	1.27

4 Discussion

This study presents a novel method that leverages AV data to estimate lane-based queue lengths and traffic signal states. This approach is particularly notable for working effectively with ultra-low penetration rates of AVs, offering a solution for managing urban traffic in a data-sparse environment.

First, the method's ability to estimate traffic signal settings demonstrated a high level of accuracy, 90% for fixed-time signals and 75% for actuated-time signals. Second, the trajectory reconstruction method leverages estimated signals and available data, even when sparse, for queue length estimation over time. The queue size mean absolute error (MAE) values of 0.67 for fixed-time signals and 1.25 for actuated-time signals at intersections suggest that our approach yields reasonably accurate queue length estimates. This result confirms that despite the challenges of working with sparse data and limited information on signal timings, our approach shows promise for accurate traffic state estimation in urban networks.

References

Chen, Xuejian, Yin, Juyuan, Tang, Keshuang, Tian, Ye, & Sun, Jian. 2022. Vehicle trajectory reconstruction at signalized intersections under connected and automated vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*, **23**(10), 17986–18000.

Guo, Qiangqiang, Li, Li, & Ban, Xuegang Jeff. 2019. Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, **101**, 313–334.

Li, Li, Jiang, Rui, He, Zhengbing, Chen, Xiqun Michael, & Zhou, Xuesong. 2020. Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C: Emerging Technologies*, **114**, 225–240.

Ramezani, Mohsen, & Geroliminis, Nikolas. 2015. Queue profile estimation in congested urban networks with probe data. *Computer-Aided Civil and Infrastructure Engineering*, **30**(6), 414–432.