A Review of Real-time Traffic State Estimation in Urban Corridors from Heterogeneous Data

Max Schrader

1 Overview

I chose to review *Real-time Traffic State Estimation in Urban Corridors from Heterogeneous Data* because of its near direct application to my research. We have recently deployed radars to a three intersection span of McFarland Boulevard in Northport, Alabama and would like to use the radars for traffic state estimation. In the short-term, the estimated traffic state will allow for the validation of simulation models. Looking towards the future, the goal is to fusion the radar data with camera feeds to get a snapshot of traffic characteristics, which is what this paper covers.

There is extensive literature on the use of Extended Kalman Filters (EKFs), Unscented Kalman Filters (UKFs) and Particle Filters (PFs) in freeway traffic state estimation. The findings of several papers point to the fact that all perform reasonably well with roughly the same results (Hegyi et al. 2006). The authors of this paper highlight that research has traditionally been on freeway state estimation, as signalized corridor state estimation is more difficult for a variety of reasons, including the fact that traffic dynamics surrounding intersections are complex and there is traditionally a lack of data at intersections. In recent years, local transportation departments have been invested in adding additional sensing capabilities to signalized intersections, will allows for sensor fusioning. The paper subject of this review relies on three sources of traffic data for state estimation: traditional loop detectors, Bluetooth tracking, and GPS devices.

The novelty of this paper is the fusioning of sensors that record at differing frequencies. The bluetooth scanners used by the authors go through a 6 second scanning cycle, the GPS data has varying frequencies, and the detector data is aggregated into 1-5 minute periods. The differences in data update frequency led the authors to develop a modified EKF algorithm that allows for the incorporation of heterogeneous measurements, whenever they become available.

2 The Dynamical System

There are several traffic models used in both practice and research. This paper focuses on a macroscopic traffic model in which individual vehicle trajectories are ignored. Specifically, it uses the LWR model written in the form

$$\frac{\partial k}{\partial t} + \frac{\partial Q(k)}{\partial s} = 0$$

where $k(s,t)$ is the traffic density function, s and t are the independent space and time variables, and $Q(k)$ is the flux function (also known as the fundamental diagram), which describes the velocity of traffic as a function of density and road parameters. The figure below illustrates the relationship between density, traffic flow, and speed (Zaidi et al. 2016).

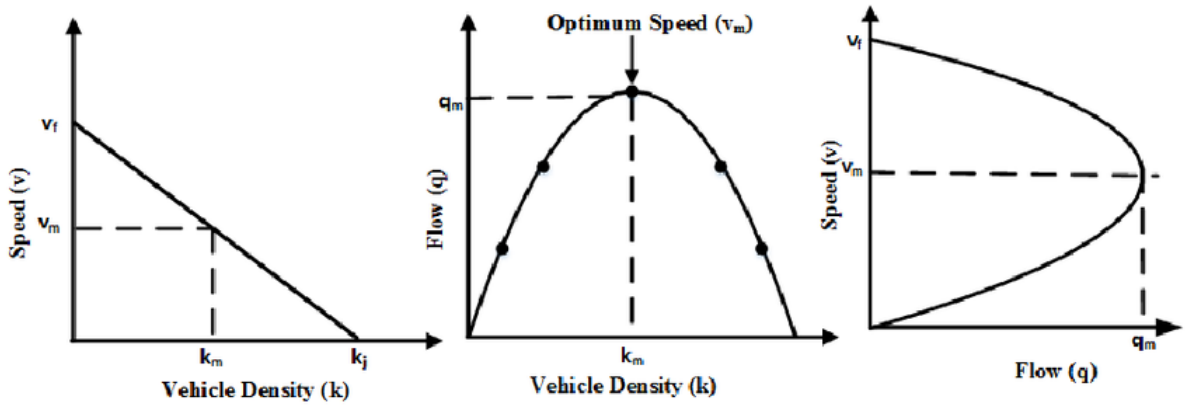

Figure 1: Greenshield's Fundamental Diagrams

The solution to the equation above is found by partitioning the corridor-of-interest into cells with size Δs and a timestep of Δt. This turns the density dynamics for a cell i at time t into:

$$k_{i,t} = k_{i,t-1} + \frac{\Delta t}{\Delta s_i}\left(q_{i,t-1}^{IN} - q_{i,t-1}^{OUT}\right)$$

where q is the numerical flow between the cells. It is capped at the capacity of vehicles that the cell can hold. To model the operation of the traffic lights in the network, a *priority ratio* is introduced, which models turning ratios and traffic light operation (when the priority ratio is set to 0, outflow from a link is stopped, emulating a red light). Figure 2 illustrates the outflow and inflow from a cell i (Nantes et al. 2016).

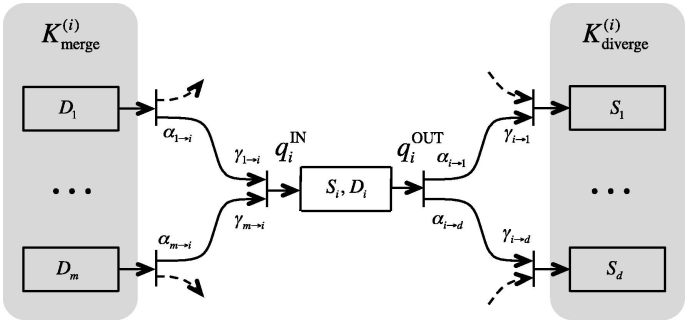

Figure 2: Illustrative Model of Macroscopic Traffic Model

3 System Model and State Estimation

The system model is used ultimately to describe the transition model, which is given succinctly as $p(x_t|x_{t-1}, u_t)$. In the case of the traffic model described above, the authors describe this transition model as

$$p\left(x_t \mid x_{t-1}, u_t\right) = x_{t-1} + F^T \underbrace{\begin{pmatrix} q_1^{IN}\left(x_{t-1}, u_t\right) - q_1^{OUT}\left(x_{t-1}, u_t\right) \\ \vdots \\ q_C^{IN}\left(x_{t-1}, u_t\right) - q_C^{OUT}\left(x_{t-1}, u_t\right) \end{pmatrix}}_{g(x_{t-1}, u_t)} + N\left(0, T\right)$$

where the full state vector, x is

$$x = \begin{pmatrix} k \\ \theta \end{pmatrix}$$

and T is

$$T = \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

where A and B are the covariance matrices of the system state variables. Some of these covariance values can be seen as *aleatory* uncertainty as they have to do with the inherent uncertainty in traffic systems, such as the number of vehicles coming in and out of a network, the exact turn ration at any given time, and individual driver characteristics.

3.1 Sensor Model's and Measurement Functions

After laying out the transition model, the authors turn to the measurement functions. In an EKF filter, the measurements are used to compute the innovation, aka $z - h$, where z is the measurement and h is the predicted state. For brevity, I will not lay out each of the measurement functions, as there are three distinct relationships between

each sensor and the system variables that they describe. In general, the sensor models can be described as

$$p\left(z_{o,t} \mid x_t, u_t \right) = h_o\left(x_t, u_t \right) + \delta_t$$

where δ_t contains the additive noise of both the system itself as well as the measurement noise. An important note is that the authors assume **conditional independence**.

The bluetooth and GPS sensors are *Lagrangian* in nature, meaning that they can capture individual vehicles' positions (speed and trajectory) at discrete moments in time. On the other hand, the induction-loop detectors are *Euclidean* in nature, as they only capture the number of vehicles that cross a point in space, without tying that measurement to any specific vehicle, thus they must be handled differently. A brief discussion of each sensor follows.

3.2 Induction Loop Detectors

The induction loop detectors are located at the intersection stop-bars in this paper, and thus measure outflow from a link. They do not provide insight into whether a vehicle goes straight, left or right, but instead simply give a time-aggregated count of vehicles. The estimated outflow of an link can be compared then to this value.

3.3 GPS

The GPS provides both position and speed. By averaging the instantaneous speeds of multiple vehicles located on the same link, the *link average speed* can be found.

3.4 Bluetooth

While the GPS data gives instantaneous position and speed data, making it relatively easy to incorporate, the Bluetooth data provides origin-destination information, which is extremely useful to traffic modelers, but less so for link-averaged values (the state parameters for this model). The bluetooth sensor reports that a unique vehicle passed location A at time t_1 and then passed location B at time time t_2. Depending on the number of links between the two sensors, this value is actually space and time averaged speed. It can be compared to the system state using the individual link average speeds. The authors use the average link speeds at time t which is selected as the midpoint between time t_1 and t_2. I don't think there is adequate discussion by the authors of how this choice interacts with the traffic light operation.

4 EKF Implementation

The EKF implementation followed what we learned in class, with a few novel (to me) additions: a *clamping function* and *incremental innovation updates*. Before presenting the EKF function pseudocode, the authors first highlight the fact that the EKF relies on linearized states. The traffic model has strong non-linearities, so it must be linearized. The authors linearize the transition function according by doing a first-order Taylor expansion, as well as the measurement function.

After the linearization, the psuedo-code implentation looks similar to our class implementation, with the exception that there was an additional for-loop, which iterates over each measurement and updates the innovation and state estimate. The number of measurements available at any given update step will vary, as the three sensors considered measure with varying frequencies. The iterative update of the Kalman gain and expected state is only possible due to the assumption of *conditional independence*.

As alluded to above, a *clamping function* is used by the authors to prevent the Kalman estimations from diverging significantly from the actual state of the system. At the end of each EKF iteration, the state estimations where forced to be within reasonable bounds of reality and the turning rations were forced to sum to 1.

5 Assumptions and Sources of Error

The authors assume that all noise and uncertainty is additive, zero-mean Gaussian noise. While this is likely a safe estimation for the sensors, there is ample traffic research that points to uniform estimation of aleatory uncertainty sources being a more accurate representation of reality (Punzo and Montanino 2020).

Another assumption is that v^{free}, k^{crit} and k^{jam} do not vary over time, which is a common assumption in macroscopic traffic simulation. In reality they do of course vary, though that can be due to difficult-to-model sources like weather and visibility.

I was underwhelmed by the authors' explanation for why they chose the EKF vs a PF or UKF. We learned in class that EKFs are the worst at dealing with non-linear systems, and thus I would have thought that a PF or UKF would preform better. Other papers point to the PF as dealing with the non-linearites of traffic the best (Chen, Rakha, and Sadek 2011).

6 Results

The proposed EKF algorithm was tested using the traffic simulation software AIMSUM. They ran 660 simulations, with varying penetration rates of GPS and Bluetooth sensing, as well as measurements with varying frequencies. They also considered varying degrees of traffic demand, including traffic jam conditions.

The method that the authors use to asses accuracy is one of the more interesting parts of the paper. Instead of comparing the actual output of the traffic simulation, they first graph the results, then compare the plots via the 2D image vectors, assessing the pixel similarities between the plots (which they call a similarity score, K). For reference, on of the diagrams is included below.

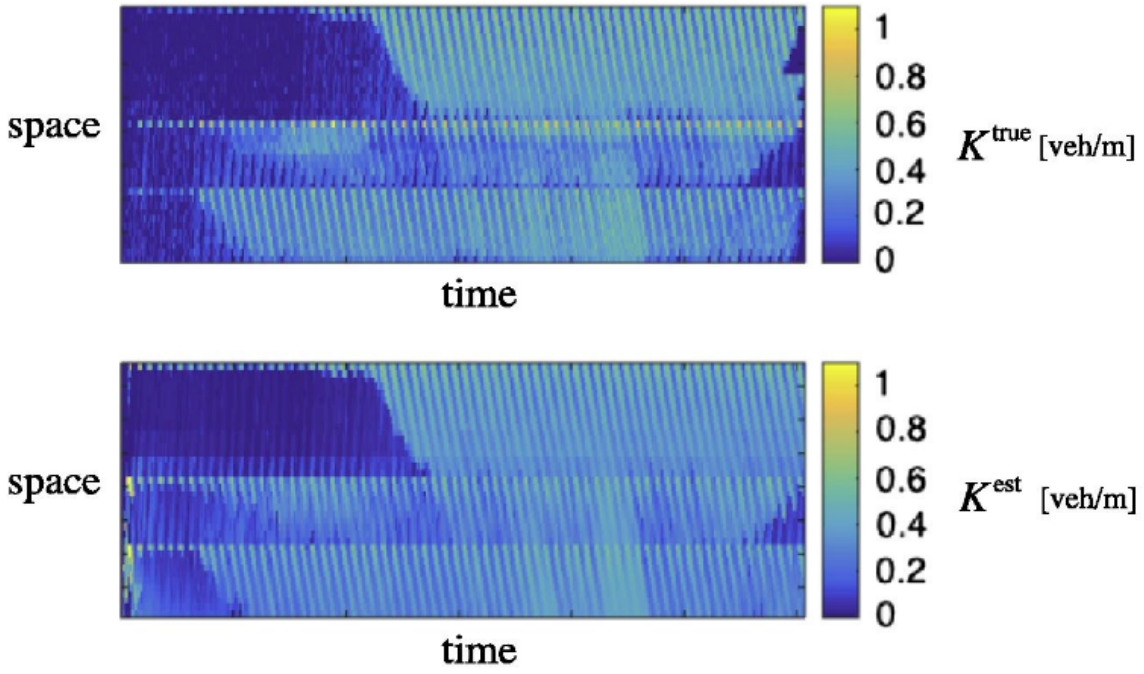

Figure 3: The spatio-temporal density diagram for one simulation run

The results show using all three sensors together leads to $K = 0.46$. I find the choice of pixel similarity interesting here, as that value provides no insight into where the EKF algorithm struggles vs where it does well.

The authors do highlight the fact that the GPS contributes the most towards the EKF accuracy. It would have been nice if they were able to plot the accuracy as a function of traffic density, as the flow characteristics become non-linear around the critical density k^{crit}.

Bibliography

10 Chen, Hao, Hesham A. Rakha, and Shereef Sadek. 2011. "Real-Time Freeway Traffic State Prediction: A Particle Filter Approach." In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 626–31. Washington, DC, USA: IEEE. doi:10.1109/ITSC.2011.6082873[1].

Hegyi, A., D. Girimonte, R. Babuska, and B. De Schutter. 2006. "A Comparison of Filter Configurations for Freeway Traffic State Estimation." In *2006 IEEE Intelligent Transportation Systems Conference*, 1029–34. doi:10.1109/ITSC.2006.1707357[2].

Nantes, Alfredo, Dong Ngoduy, Ashish Bhaskar, Marc Miska, and Edward Chung. 2016. "Real-Time Traffic State Estimation in Urban Corridors from Heterogeneous Data." *Transportation Research Part C: Emerging Technologies* 66 (May): 99–118. doi:10.1016/j.trc.2015.07.005[3].

Punzo, Vincenzo, and Marcello Montanino. 2020. "A Two-Level Probabilistic Approach for Validation of Stochastic Traffic Simulations: Impact of Drivers' Heterogeneity Models." *Transportation Research Part C: Emerging Technologies* 121 (December): 102843. doi:10.1016/j.trc.2020.102843[4].

Zaidi, Kamran, Milos B. Milojevic, Veselin Rakocevic, Arumugam Nallanathan, and Muttukrishnan Rajarajan. 2016. "Host-Based Intrusion Detection for VANETs: A Statistical Approach to Rogue Node Detection." *IEEE Transactions on Vehicular Technology* 65 (8): 6703–14. doi:10.1109/TVT.2015.2480244[5].

[1] https://doi.org/10.1109/ITSC.2011.6082873
[2] https://doi.org/10.1109/ITSC.2006.1707357
[3] https://doi.org/10.1016/j.trc.2015.07.005
[4] https://doi.org/10.1016/j.trc.2020.102843
[5] https://doi.org/10.1109/TVT.2015.2480244