

Assimilating Eulerian and Lagrangian data in traffic-flow models

Chao Xia* Courtney Cochran[†] Joseph DeGuire[‡] Gaoyang Fan[§] Emma Holmes[¶]
Melissa McGuirl^{||} Patrick Murphy[§] Jenna Palmer^{**} Paul Carter^{††} Laura Slivinski^{‡‡}
Björn Sandstede^{||}

November 9, 2016

Abstract

Data assimilation of traffic flow remains a challenging problem. One difficulty is that data come from different sources ranging from stationary sensors and camera data to GPS and cell phone data from moving cars. Sensors and cameras give information about traffic density, while GPS data provide information about the positions and velocities of individual cars. Previous methods for assimilating Lagrangian data collected from individual cars relied on specific properties of the underlying computational model or its reformulation in Lagrangian coordinates. These approaches make it hard to assimilate both Eulerian density and Lagrangian positional data simultaneously. In this paper, we propose an alternative approach that allows us to assimilate both Eulerian and Lagrangian data. We show that the proposed algorithm is accurate and works well in different traffic scenarios and regardless of whether ensemble Kalman or particle filters are used. We also show that the algorithm is capable of estimating parameters and assimilating real traffic observations and synthetic observations obtained from microscopic models.

1 Introduction

Estimating the traffic state on a network of highways is a challenging problem. With the availability of stationary sensor data, traffic cameras, and GPS cell phone data, more and more data can now be used to predict car density, velocity, and travel time in real time. Predicting traffic involves (i) a model that provides the traffic at later times based on given initial conditions and (ii) techniques to incorporate and assimilate actual traffic-state observations into the initial conditions to improve the model forecast. To formulate a framework for estimating traffic flow, we therefore need a mathematical model of traffic flow, a sense of what types of observations are available, and a scheme for assimilating these observations into the model. We now discuss these three ingredients in turn.

Mathematical models for traffic flow come in many different flavors. Common models range from cellular automata or microscopic car-following models for individual cars to macroscopic discrete or partial-differential equation (PDE) models for car densities, velocities, and possibly other quantities [20, 25, 32]. Knowledge of the positions and velocities of all cars on a road or road network is required to successfully employ a microscopic model. Since most of this information is not well known, it is typically not feasible to use such models for traffic flow prediction. Thus, macroscopic models are used more frequently for traffic estimation. A commonly

*Corresponding author: Chao Xia

E-mail address: chao.xia@brown.edu Tel.: +1 401 863 2335 Fax: +1 401 863-1355

Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

[†]Department of Mathematics, Davidson College, Davidson, NC 28035, USA

[‡]Department of Economics and Mathematics, University of Wisconsin, Madison, WI 53706, USA

[§]Department of Mathematics, University of Utah, Salt Lake City, UT 84112, USA

[¶]Department of Mathematics, Colorado College, Colorado Springs, CO 80903, USA

^{||}Division of Applied Mathematics, Brown University, Providence, RI 02912, USA

^{**}Department of Earth, Environmental and Planetary Science, Brown University, Providence, RI 02912, USA

^{††}Department of Mathematics, Brown University, Providence, RI 02912, USA

^{‡‡}Cooperative Institute for Research in Environmental Sciences, University of Colorado Boulder, Boulder, CO 80309, USA

used minimal macroscopic PDE model is the Lighthill–Whitham equation [25]. In this paper, we will use variants of the Lighthill–Whitham equation and assume that the velocity and density are related linearly via the Greenshields function. This model will be discussed in more detail in §2.1 below.

Traffic-flow observations come from a variety of sources and are available as functions of time. Stationary sensors, such as induction loops or cameras, provide the flux, average velocity, and local density of cars that move past the fixed sensor location. GPS data from cell phones or navigation devices, on the other hand, provide information about the positions and velocities of individual cars that move with the traffic flow. We refer to observations that come from a fixed observation location as Eulerian observations and to observations that come from parcels (cars) that move with the traffic flow as Lagrangian observations.

There are a number of different data-assimilation techniques available that help incorporate traffic observations at discrete times into an underlying model. Most of these are based on a Bayesian statistics analysis that treats the forecast from the model as the prior distribution and then calculates a posterior distribution based on the available observations. Commonly, traffic estimators employ ensemble Kalman filters for this calculation. The main limitation of the existing data-assimilation techniques stems from the fact that sensor and GPS data require very different assimilation approaches: it is therefore technically involved to use existing techniques to assimilate Eulerian and Lagrangian observations simultaneously.

For instance, if we use a partial differential equation for the car density $\rho(x, t)$ at position x and time t as our underlying model, then sensor data produce estimates for the car density $\rho(x_0, t)$ at the sensor location x_0 , which can be compared directly to the prediction made by the model. GPS data, on the other hand, are Lagrangian in nature, that is, they provide the location $x(t)$ and velocity $v(t)$ of an individual car as it moves with the traffic but cannot produce an estimate of the car density at that moving location. In particular, such data cannot be assimilated directly into a density-based model.

Two approaches have been considered to accomplish the assimilation of Lagrangian data. First, the underlying model can be written in terms of Lagrangian coordinates [41, 42] that move with the speed of the underlying cars, which makes it possible to assimilate Lagrangian observations. However, it is then much more difficult to assimilate Eulerian observations (such as sensor data); moreover, the approach outlined in [41, 42] was applied only to certain types of Eulerian observations (namely the velocities of cars measured at fixed sensor locations, but not the densities or fluxes recorded at these fixed locations). Second, one can convert the partial differential equation for the density to an equation for the velocity by inverting the density-velocity function $V(\rho)$ and then assimilate velocity observations [39, 40]; however, this inversion may not be possible for other models.

The main contribution of this paper is to propose a method that provides an efficient alternative approach to the assimilation of Lagrangian GPS data, whilst keeping the ability to simultaneously assimilate Eulerian sensor data. The idea behind the proposed method is to append the differential equations for the positions of the cars for which GPS data are available to the macroscopic model to solve simultaneously for the positions of these cars. This approach was originally developed in the context of assimilating data from drifters and floaters in fluid flows [28, 33]. The advantage of this method compared to the previous approaches outlined above is that it does not require any reformulation of the underlying macroscopic model. We show here that this technique can be used efficiently for traffic flow.

To explain this approach in more detail, consider, for instance, the Lighthill–Whitham equation

$$\rho_t + (\rho V(\rho))_x = 0, \quad (1)$$

where $\rho(x, t)$ denotes the density of cars at position x and time t , and $V = V(\rho)$ is a constitutive law that relates density and velocity. If $p_j(t)$ with $j = 1, \dots, N_c$ denotes the position of the j th car that provides GPS data, then the closed system

$$\rho_t(x, t) + [\rho(x, t)V(\rho(x, t))]_x = 0, \quad \frac{dp_j}{dt} = V(\rho(p_j, t)), \quad j = 1, \dots, N_c \quad (2)$$

describes the density $\rho(x, t)$ and the positions $p_j(t)$ for all (x, t) . We can now assimilate position and velocity data of the N_c cars as well as sensor data at the N_s fixed locations q_i for $i = 1, \dots, N_s$ through the linear observation operator

$$(\rho, (p_j)_{j=1, \dots, N_c}) \mapsto ((\rho(q_i, t))_{i=1, \dots, N_s}, (p_j(t))_{j=1, \dots, N_c}).$$

Computationally, the only additional expense is to solve the N_c ordinary differential equations (2).

In terms of filter techniques for assimilating observations, we will employ both ensemble Kalman filters and particle filters, which will be discussed in more detail below. We remark that data assimilation has also been used to estimate parameters ([5, 26]), which may, for instance, appear in the constitutive law $V(\rho)$, as these may not be known a priori and can also change over time.

In this paper, we will discuss the efficacy and accuracy of the proposed method for assimilating sensor and GPS data in traffic flow. Specifically, we will demonstrate that the proposed algorithm is robust with respect to different traffic scenarios and its implementation using ensemble Kalman filters or particle filters. Thus, our method provides a viable alternative to existing data assimilation techniques in their ability to assimilate both Eulerian and Lagrangian data. We will also show that the algorithm is capable of estimating the typically unknown parameters that appear in the underlying macroscopic models. Observations coming from microscopic traffic flow models and from real data will be used to show that the algorithm works well when the observations are not generated from the underlying model (1), provided the traffic flow is compatible with the computational model that is used to assimilate the data.

The remainder of this paper is organized as follows. In §2, we outline our algorithms for assimilating Eulerian sensor data and Lagrangian GPS data into macroscopic traffic models; we also briefly review the data assimilation methods, and discuss localization, inflation, resampling and parameter estimation. Our main results are presented in §3 and §4, and we conclude with a discussion of our results and open problems in §5.

2 Data assimilation of Eulerian and Lagrangian data in traffic-flow models

In this section, we describe our scheme for assimilating Eulerian and Lagrangian data into macroscopic traffic-flow models. Our approach will allow us to simultaneously assimilate stationary camera and sensor data as well as GPS data from moving cars.

2.1 Underlying macroscopic traffic model: Lighthill–Whitham

We use the Lighthill–Whitham equation as the underlying core macroscopic mathematical model of traffic flow. In this model, the traffic is described by the vehicle density $\rho(x, t)$ at location x and time t . For simplicity, we consider a ring road of length L but emphasize that roads with other boundary conditions can also be treated by our approach. In the absence of sinks and sources, and adding a diffusion term with diffusion coefficient $\epsilon > 0$ to account for low-level noise, the conservation law for the vehicle density $\rho(x, t)$ with periodic boundary conditions is given by

$$\begin{aligned}\partial_t \rho(x, t) + \partial_x \phi(\rho(x, t)) &= \epsilon \partial_x^2 \rho(x, t), & x \in \mathbb{T}, \\ \rho(x, 0) &= \rho_0(x),\end{aligned}\tag{3}$$

where $\rho_0(x)$ denotes the initial data, and \mathbb{T} denotes the circle of circumference L that reflects the periodic boundary conditions we employ. The flux function $\phi(\rho)$ expresses the dependence of the vehicle flux ϕ on the density ρ ; this relationship is usually referred to as the fundamental diagram. We assume that ϕ is defined on the interval $[0, \rho_{\max}]$, where ρ_{\max} is the maximal density achievable on the road.

For traffic flow, we can write $\phi(\rho) = \rho V(\rho)$, where the function $V(\rho)$ relates velocity v and density ρ for densities $\rho \in [0, \rho_{\max}]$. The function $V(\rho)$ is a modeling choice: examples include the Daganzo–Newell velocity function [39] and the Greenshields affine velocity function [17]. In this paper, we use the linear velocity function given by Greenshields:

$$V_G(\rho) = v_{\max} \left(1 - \frac{\rho}{\rho_{\max}} \right),\tag{4}$$

where v_{\max} is the maximal free-flow velocity. Traffic jams manifest themselves as shock waves in (3), see Figure 1, which travel with characteristic speed c , where positive speeds correspond to propagation of traffic jams against the traffic flow.

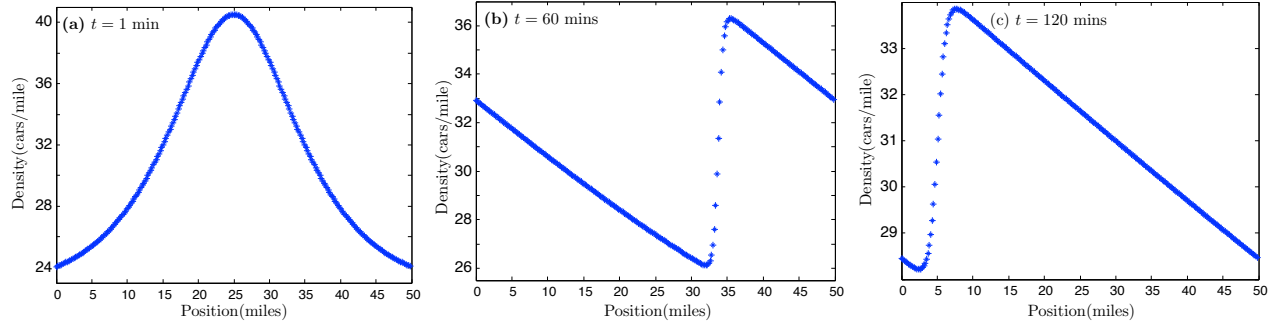


Figure 1: *Evolution into a shock in the viscous Lighthill-Whitham equation (3). The shock is moving backward with gradually decreasing amplitude.*

Given a solution $\rho(x, t)$ of (3), we can recover the movement of individual cars. Indeed, if $(p_i(t))_{i=1,2,\dots,N_c}$ denote the positions of N_c individual cars as functions of time, then these functions satisfy the ordinary differential equations

$$\frac{dp_i}{dt} = V(\rho(p_i, t)), \quad i = 1, 2, \dots, N_c,$$

which can be solved to provide car positions for all times given their initial locations at $t = 0$.

The macroscopic model (3) can be used to describe normal traffic flow in the absence of ramps, traffic lights, and construction zones. In §3 below, we will add additional terms to the model (3) to capture the effects of on- and off-ramps, traffic lights, road construction sites, and traveling bottlenecks. The modifications allow us to simulate common traffic scenarios and therefore enable us to better test and validate the data assimilation strategies proposed here. In the remainder of this section, we will focus on (3) but stress that the techniques outlined in this section will also apply to the modifications of (3) discussed in §3 below.

2.2 Eulerian and Lagrangian observations

There are two main categories of observation models in traffic state estimation: one is Eulerian observation in which the data is collected from stationary sensors on the road such as cameras, induction loops, and radar; the other type is Lagrangian observation, which includes the position and velocity data from moving vehicles equipped with GPS devices or cell phones.

2.2.1 Continuous observation models

Eulerian observation data: Sensors deployed at fixed positions along roads are capable of counting the number of vehicles passing by as well as reporting the flux at their location as a function of time. Thus, if there are N_s sensors positioned at the fixed locations $(q_i)_{i=1,2,\dots,N_s}$ on a ring road \mathbb{T} , then the complete system of traffic model (M) and Eulerian observation operators (D) is given by

$$\begin{aligned} \text{(M)} \quad & \rho_t(x, t) + \partial_x \phi(\rho(x, t)) = \epsilon \rho_{xx}(x, t), \quad x \in \mathbb{T} \\ \text{(D)} \quad & h^\phi(\rho) := (\phi(\rho(q_i, t)))_{i=1,2,\dots,N_s}. \end{aligned} \tag{5}$$

An advantage of Eulerian sensor data is their high accuracy: modern traffic sensors are so sensitive that the probability of missing or over-counting a vehicle is below 0.1% ([1]). However, the Eulerian data is expensive to collect, and fixed locations of sensors limit the observation coverage. Note also that the Eulerian observation function $h^\phi(\cdot)$ maps the vehicle density ρ to the vehicle flux $\phi(\rho)$ and is therefore typically nonlinear: the nonlinearity of the observation function will influence the efficacy of data assimilation, especially for the ensemble Kalman filter, and requires special treatment which will be discussed in §2.3.1.

Lagrangian observation data: Lagrangian observations consist typically of the positions and velocities of moving vehicles collected by GPS devices or cell phones. If there are N_c cars equipped with these devices, and their positions and velocities are denoted by $(p, v) := (p_j, v_j)_{j=1,2,\dots,N_c}$, then the complete system of the traffic model (M), capable of tracking the macroscopic traffic density and the positions of the N_c individual cars, and the Lagrangian observation operators (D) is given by

$$\begin{aligned} \text{(M)} \quad & \begin{cases} \rho_t(x, t) + \partial_x \phi(\rho(x, t)) = \epsilon \rho_{xx}(x, t), & x \in \mathbb{T} \\ \dot{p}_j(t) = v_j(t) = V(\rho(p_j(t), t)), & j = 1, 2, \dots, N_c \end{cases} \\ \text{(D)} \quad & \begin{cases} h^p(p, v) = p \\ h^v(p, v) = v. \end{cases} \end{aligned} \quad (6)$$

Thus, in contrast to previous methods for assimilating Lagrangian data that relied on its reformulation in Lagrangian coordinates or reshaping the macroscopic traffic model in terms of velocity or spacing [41, 42], our approach adds the governing equations for the positions of individual cars to the macroscopic model by exploiting that our constitutive law relates density and velocity.

The main advantage of Lagrangian data is the broad coverage of traffic states as the tracked vehicles travel along the road. The disadvantage is that density information is not collected directly. Furthermore, the observation noise of Lagrangian data is relatively large (see below for details) and depends, for instance, on the carrier's signal and weather conditions.

Eulerian and Lagrangian observation data: We now outline how Eulerian and Lagrangian data can be assimilated simultaneously by concatenating the systems (5) and (6) that we introduced above. The resulting system for the traffic flow model (M) and the observation operator (D) is given by

$$\begin{aligned} \text{(M)} \quad & \begin{bmatrix} \rho_t(x, t) \\ \dot{p}_j(t) \\ v_j(t) \end{bmatrix} = \begin{bmatrix} \epsilon \rho_{xx}(x, t) - \partial_x \phi(\rho(x, t)) \\ V(\rho(p_j(t), t)) \\ V(\rho(p_j(t), t)) \end{bmatrix} \quad \begin{matrix} j = 1, 2, \dots, N_c \\ j = 1, 2, \dots, N_c \end{matrix} \\ \text{(D)} \quad & \begin{bmatrix} h^\phi(\rho) \\ h^p(p, v) \\ h^v(p, v) \end{bmatrix} = \begin{bmatrix} (\phi(\rho(q_i, t)))_{i=1,2,\dots,N_s} \\ p \\ v \end{bmatrix}. \end{aligned} \quad (7)$$

The framework proposed in (7) is capable of assimilating both Eulerian sensor and Lagrangian GPS data as it is a closed system for the density ρ and the locations and velocities of the N_c vehicles from which GPS data are gathered. We note that we will consider N_c to be fixed but emphasize that our framework allows this quantity to depend on function of time as long as it changes slowly compared to the time scale over which GPS data are collected.

Observation errors: We assume that the observation errors for sensor, GPS position, and GPS velocity data are independent and normally distributed with zero mean. As mentioned above, the observation noise η_i^ϕ for sensor data has a variance of $0.1\% \phi(q_i, t)$. Based on the current accuracy of GPS data, we assume that position and velocity measurements have errors with standard deviation of 5.12m ([16]) and 0.0707m/s ([29, 31]), respectively. Using these assumptions, the covariance matrix of the observation errors can be written as

$$\mathbf{R}(t) = \begin{bmatrix} 0.1\% (\phi(q_i, t))_{i=1,\dots,N_s} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & (5.12\text{m})^2 \mathbb{I}_{N_c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (0.0707\text{m/s})^2 \mathbb{I}_{N_c} \end{bmatrix}, \quad (8)$$

where \mathbb{I}_N denotes the $N \times N$ identity matrix.

2.2.2 Discretized observation models

We now discuss the numerical implementation of the continuous scheme (7), which is obtained by discretization in space and time. Recall that we consider a ring road \mathbb{T} of length L .

First, we consider the discretization of the traffic model (M) given by

$$\begin{bmatrix} \rho_t(x, t) \\ \dot{p}_j(t) \\ v_j(t) \end{bmatrix} = \begin{bmatrix} \epsilon \rho_{xx}(x, t) - \partial_x \phi(\rho(x, t)) \\ V(\rho(p_j(t), t)) \\ V(\rho(p_j(t), t)) \end{bmatrix} \quad \begin{matrix} j = 1, 2, \dots, N_c \\ j = 1, 2, \dots, N_c \end{matrix} \quad (9)$$

For the spatial discretization, we pick a large integer M , define a spatial step size $\Delta x := L/M$, and choose M consecutive mesh points $x_m \in \mathbb{T}$ that are equally spaced along the ring road with distance Δx . Time is discretized by choosing a small positive time step ΔT and evaluating solutions at times $t = n\Delta T$ for integers $n \in \mathbb{N}$.

The traffic state at time $t = n\Delta T$ consists of the densities at each mesh point $x_m \in \mathbb{T}$ as well as the GPS positions and velocities of tracked vehicles and is therefore given by

$$\mathbf{X}(t) := [\underbrace{\rho(x_1, t), \dots, \rho(x_M, t)}_{\text{densities}}, \underbrace{p_1(t), \dots, p_{N_c}(t)}_{\text{GPS positions}}, \underbrace{v_1(t), \dots, v_{N_c}(t)}_{\text{GPS velocities}}]^T. \quad (10)$$

The discretized system obtained from solving (9) numerically can then be written as

$$\mathbf{X}(t) = \mathbf{f}(\mathbf{X}(t - \Delta T); \boldsymbol{\theta}(t - \Delta T)), \quad (11)$$

where $\boldsymbol{\theta}(t)$ denotes system parameters, which may be constant or vary in time. The function \mathbf{f} is obtained using the following numerical schemes. First, we discuss how we discretize the viscous Lighthill–Whitham equation for the density $\rho(x, t)$ that appears in (9): for periodic boundary conditions, we approximate the spatial derivatives in the viscous Lighthill–Whitham equation for the density $\rho(x, t)$ using spectral differentiation matrices in physical space as outlined in [37, (3.10) in §3]; for non-periodic boundary conditions, we discretize the spatial derivatives using finite differences. Next, we describe how we evaluate the equations for the positions and velocities (p_j, v_j) that appear in (9): to determine the values $\rho(p_j, t)$ of the density at arbitrary positions p_j , we interpolate ρ from its known values $\rho(x_m, t)$ at the mesh points x_m using trigonometric interpolation ([21]). Finally, we integrate the resulting system of differential equations from time $t - \Delta T$ to time t using a third-order Runge–Kutta method.

We now move to the observations. Using the observation errors introduced above, Eulerian and Lagrangian observations can be written as

$$\mathbf{Y}(t) = \mathbf{h}(\mathbf{X}(t)) + \boldsymbol{\eta}(t), \quad \boldsymbol{\eta}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(t)),$$

where $\mathbf{h}(\mathbf{X}(t))$ with $\mathbf{X}(t)$ given in (10) is defined via

$$\mathbf{h}(\mathbf{X}(t)) := (\phi(\rho(q_1, t)), \dots, \phi(\rho(q_{N_s}, t)), p_1(t), \dots, p_{N_c}(t), v_1(t), \dots, v_{N_c}(t))^T. \quad (12)$$

In order to evaluate $\rho(q_i, t)$ at the location q_i of a sensor, we can either arrange that q_i coincides with a mesh point x_m or again use trigonometric interpolation ([21]). We remark that the expression in (12) can be modified easily to account for the situation where we have observations from sensors only or from GPS data only available for data assimilation.

2.3 Ensemble Kalman and particle filters

We now turn to data assimilation and describe how filters can be used to estimate traffic states and uncertain parameters in the traffic flow models described in §2.1. We first briefly review the ensemble Kalman filter (EnKF) and the particle filter (PF), which are two common sequential data assimilation methods. These filters estimate the Bayesian prior and posterior probability distribution using an ensemble of possible states. Bayes' rule gives the true posterior distribution of the traffic state \mathbf{X} given the observations \mathbf{Y} as

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})}. \quad (13)$$

In both the EnKF and PF methods, the distributions appearing in Bayes' rule are approximated by a weighted ensemble of the state \mathbf{X} given by $\{\mathbf{X}_i, w_i\}_{i=1}^{N_c}$. We will use the notation that $\{\mathbf{X}_i^f, w_i^f\}_{i=1}^{N_c}$ is the ensemble from the prior distribution $p(\mathbf{X})$ whereas $\{\mathbf{X}_i^a, w_i^a\}_{i=1}^{N_c}$ is from the posterior distribution $p(\mathbf{X}|\mathbf{Y})$.

2.3.1 Ensemble Kalman filter

The ensemble members are equally weighted for the entire assimilation window and are updated according to an ensemble approximation of the traditional filter update step, given here by the so-called perturbed observation EnKF [8, 12, 22]

$$\mathbf{X}_i^a = \mathbf{X}_i^f + \mathbf{K}(\mathbf{Y} - \mathbf{H}\mathbf{X}_i^f + \boldsymbol{\eta}_i), \quad (14)$$

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1}, \quad (15)$$

where \mathbf{X}_i^f is the forecast of the i^{th} ensemble member, \mathbf{X}_i^a is the i^{th} updated (analysis) ensemble member, \mathbf{H} is the observation operator, \mathbf{K} is the Kalman gain matrix, \mathbf{R} is the covariance of the observation error, and $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \mathbf{R})$ are the observation perturbations. \mathbf{P}^f is the forecast ensemble covariance given by

$$\mathbf{P}^f = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\mathbf{X}_i^f - \bar{\mathbf{X}}^f)(\mathbf{X}_i^f - \bar{\mathbf{X}}^f)^T, \quad \bar{\mathbf{X}}^f = \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{X}_i^f. \quad (16)$$

Nonlinear observation function: We note that the update step requires that the observation operator \mathbf{H} is linear. However, the observation function given in (12) is nonlinear, and we will therefore outline how we adjust the update step to accommodate the nonlinear observation function. Instead of calculating the operator \mathbf{H} and the covariance matrix \mathbf{P}^f separately when estimating the state \mathbf{X}_i^a in (14) and (15), we follow [27] and treat $\mathbf{P}^f \mathbf{H}^T$ and $\mathbf{H} \mathbf{P}^f \mathbf{H}^T$ as follows as single entities:

$$\mathbf{H} \mathbf{P}^f \mathbf{H}^T = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\mathbf{Z}_i^f - \bar{\mathbf{Z}}^f)(\mathbf{Z}_i^f - \bar{\mathbf{Z}}^f)^T, \quad \bar{\mathbf{Z}}^f := \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{Z}_i^f, \quad (17)$$

$$\mathbf{P}^f \mathbf{H}^T = \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\mathbf{X}_i^f - \bar{\mathbf{X}}^f)(\mathbf{Z}_i^f - \bar{\mathbf{Z}}^f)^T, \quad \bar{\mathbf{X}}^f := \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{X}_i^f, \quad (18)$$

where $\mathbf{Z}_i^f := h(\mathbf{X}_i^f)$ denotes the ensembles projected into observation space. Similarly, the term $\mathbf{H}\mathbf{X}_i^f$ in (14) is replaced by \mathbf{Z}_i^f .

Localization: The EnKF can perform well in high-dimensional systems when covariance localization and inflation are used [4, 18, 22, 23]. When the ensemble size is much smaller than the dimension of the state, spurious correlations can arise in the sample covariance matrix. These spatially long-range correlations are diminished through the method of localization.

One common localization method applies an exponential decay function, such as the Gaspari–Cohn function in [14], to the forecast covariance matrix \mathbf{P}^f . In our formulation of the Kalman gain matrix, the forecast covariance matrix \mathbf{P}^f is not directly accessible as we provide only $\mathbf{H} \mathbf{P}^f \mathbf{H}^T$ and $\mathbf{P}^f \mathbf{H}^T$ through (17) and (18), respectively. As in [23], we could now apply elementwise localization separately to each of these two matrices: localization of $\mathbf{P}^f \mathbf{H}^T$ would account for the assumption that observations at a given location would only affect the density and car positions near that location, while localization of $\mathbf{H} \mathbf{P}^f \mathbf{H}^T$ would correspond to localizing the effect observations at different positions have on each other.

Alternatively, if we interpret the operator $\mathbf{K} = (\mathbf{K}_{ij})$ loosely as a weight matrix where the (i, j) th entry \mathbf{K}_{ij} determines how much the j th entry of the observation vector affects the i th entry of the state vector, then we can apply localization directly to the matrix \mathbf{K} to ensure that observations at a certain location affect only nearby states. Specifically, we assume that only mesh points within 0.5 miles of the location of a sensor are affected by Eulerian observations. We then define an exponential decay localization function for the j th column of the Kalman gain matrix \mathbf{K} via

$$e^{-d|x_m - (q_j + s)|} \quad \text{with} \quad |x_m - q_j| < 0.5 \text{ miles},$$

where x_m is the coordinate of the m th mesh point, q_j is the coordinate of the j th sensor, and d is a decay coefficient that describes the decay rate of the localization function. Since cars detected by a sensor will have

travelled a certain distance s during time step ΔT , we center the localization at $q_j + s$. Throughout, we use the parameters $d = 0.5$ and $s = 0.35$ miles. We use the same localization function also for Lagrangian observations: we set $d = 1.2$ and choose $s = 0$ as there will be no time delay for positions and velocities collected from GPS data.

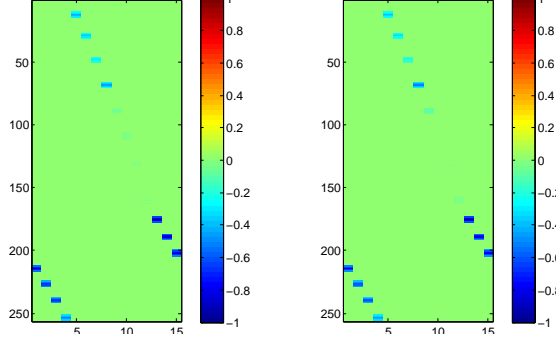


Figure 2: Comparison of the scaled Kalman gain matrices with localization applied to the forecast covariance matrix (left) and directly to the Kalman gain matrix (right).

Figure 2 shows that applying localization directly to the Kalman gain matrix turns out to be very similar to applying localization to the forecast covariance matrix. Indeed, our numerical results show that the observation covariance $\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R}$ is dominated by its diagonal elements: using the notation \mathbf{I} for the identity matrix, we then see that the Kalman gain matrix, with localization applied to the forecast covariance matrix, can be approximated by the Kalman gain matrix with direct localization:

$$\begin{aligned} \left(\mathbf{loc} \circ (\mathbf{P}^f \mathbf{H}^T) \right) \left(\mathbf{I} \circ (\mathbf{H} \mathbf{P}^f \mathbf{H}^T) + \mathbf{R} \right)^{-1} &= \mathbf{loc} \circ \left(\mathbf{P}^f \mathbf{H}^T \left(\mathbf{I} \circ (\mathbf{H} \mathbf{P}^f \mathbf{H}^T) + \mathbf{R} \right)^{-1} \right) \\ &\approx \mathbf{loc} \circ \left(\mathbf{P}^f \mathbf{H}^T \left(\mathbf{H} \mathbf{P}^f \mathbf{H}^T + \mathbf{R} \right)^{-1} \right). \end{aligned}$$

Inflation: The EnKF is not immune to filter divergence, where the prior ensemble spans a smaller and smaller space until the observations no longer have any effect in the analysis step ([4]). The EnKF is prone to artificial collapse of the covariance matrix. To address this issue, the covariance may be “inflated” before the Kalman update step is performed. We use two different inflation algorithms for the EnKF. First, we employ a constant scalar inflation for the simulations of multiple traffic scenarios as the localization algorithm described above does not significantly tighten the covariance matrix. Second, for the computations presented in §4.2, we use adaptive inflation as described in [2, 3].

2.3.2 Particle filter

The posterior is approximated by updating the weights but leaving the particle positions fixed so that $\mathbf{X}_i^a = \mathbf{X}_i^f =: \mathbf{X}_i$. The updated weights are obtained by applying Bayes’ rule to the weights as follows:

$$w_i^a = \frac{p(\mathbf{Y}|\mathbf{X}_i)w_i^f}{\sum_{j=1}^{N_e} p(\mathbf{Y}|\mathbf{X}_j)w_j^f}, \quad (19)$$

that is, the updated weights are found by multiplying the likelihood of that particle by the previous weight and normalizing to sum to one ([11, 15]). For a more complete derivation of the particle filter and the proposal distributions, see [10, 34, 36].

To prevent the weights of the ensemble concentrating on a single state, various resampling methods for the ensemble states may be used ([38]). The basic idea behind each resampling method is to monitor when a predetermined threshold is hit (for example, when the “effective sample size” becomes small ([24])), and to then

resample the particles from the discrete approximation of the posterior distribution and reset all weights to $1/N_e$. In this paper, we approximate the effective sample size (following ([24])) to be

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^{N_e} w_i^2} = \begin{cases} N_e & \text{for } w_j = 1/N_e \\ 1 & \text{for } w_j = \delta_{ij} \end{cases}. \quad (20)$$

We then apply the Markov Chain Monte Carlo Metropolis-Hastings method [19] to the PF, where we set the predetermined threshold to 5% of the total particle number N_e as in (20), i.e. we pick $N_{\text{eff}}^{\text{thresh}} = 0.05N_e$. We also add a small amount of noise to each particle after resampling in order to spread them out, which is similar to the inflation process in the EnKF.

One major drawback of the traditional implementation of the PF is that it has been shown to fail in high dimensions ([35]). However, when the state dimension is small enough and the number of particles is large enough, the PF can provide an accurate approximation to the exact Bayesian posterior distribution, in contrast to the EnKF. This is especially useful if this distribution is skewed or multimodal, which is often the case when the dynamical system governing the state is nonlinear.

2.4 Parameter estimation

The macroscopic model we utilize may contain parameters whose values are not known: for instance, the expression (4) for the Greenshields velocity function contains the parameters v_{max} and ρ_{max} , which will generally be unknown.

We use data assimilation to estimate unknown parameters in addition to the traffic states by appending the unknown parameters to the state vector ([5, 26]) and updating parameters and traffic states concurrently. To model the evolution of parameters between assimilation steps, we use the artificial evolution of parameters suggested in [26], in which the parameter evolution model is given by

$$\begin{aligned} \boldsymbol{\theta}(t) &= \boldsymbol{\theta}(t - \Delta T) + \boldsymbol{\xi}(t), \\ \boldsymbol{\xi}(t) &\sim \mathcal{N}(0, \mathbf{W}_{t+1}), \end{aligned}$$

where \mathbf{W}_{t+1} is a specified covariance matrix.

2.5 Performance criteria

In this paper, we use both absolute and relative root-mean-square-errors (RMSE) as performance indicators to assess different scenarios. The absolute performance index at time t_n is defined as

$$\text{RMSE}_A = \sqrt{\frac{\sum_{m=1}^M (\rho_m(t_n) - \hat{\rho}_m(t_n))^2}{M}}, \quad (21)$$

while the corresponding relative performance index is given by

$$\text{RMSE}_R = \frac{\text{RMSE}_A}{\rho_{\text{max}}}, \quad (22)$$

where ρ_{max} is the maximal density in the macroscopic model. Here, M is the number of mesh points used to discretize a given highway stretch, $\rho_m(t_n)$ denotes the true density at the m th mesh point x_m at time t_n , and $\hat{\rho}_m(t_n)$ denotes the corresponding estimation from data assimilation. For those traffic scenarios with unknown maximal density ρ_{max} , the denominator in the relative RMSE is replaced by the average $\sum_{m=1}^M \rho_m(t_n)/M$ of the true density.

3 Data assimilation results: synthetic truth generated from the underlying model

The main goal of this section is to apply data assimilation to estimate traffic states and parameters using the theory described in the previous sections. Specifically, we will investigate

1. assimilation of Eulerian and Lagrangian data in dynamic traffic flow;
2. efficacy of Lagrangian data assimilation;
3. impact of sensor location on data assimilation;
4. parameter estimation of traffic flow models.

We note that the first three topics are investigated under the assumption that the estimator has the exact values of model parameters, and the emphasis is placed on the capability of traffic state estimators. However, in the last topic, the parameter estimation is activated with the intent of investigating the capability of parameter estimators.

3.1 Basic setup

In the simulations outlined in §3.2–3.5, the underlying model for the data assimilation is the same as the model generating the truth; in each case, the viscous Lighthill-Whitham equation (3) is used with additional terms to simulate various traffic scenarios, to be outlined individually below. We will show that data assimilation can be used to reduce the error in simulating these scenarios when imperfect initial data is used. The initial data for density is set as the truth plus 10% noise through Fourier coefficients of the density curve.

For each traffic scenario and associated macroscopic system, we consider traffic flow on a ring road of length $L = 50$ miles, which is equally divided into 256 mesh points for the numerical computation. The truth is generated by initializing the system with the profile

$$\rho_0(x) = 0.5\rho_{\max} + 0.4\rho_{\max}\text{sech}(x - L/2), \quad (23)$$

and the parameters

$$\rho_{\max} = 45 \text{ cars/mile}, \quad v_{\max} = 75 \text{ miles/hour}, \quad \epsilon = 0.1. \quad (24)$$

For the data assimilation algorithm, the underlying model is taken to be the same as that generating the synthetic truth; however, the initial conditions for the ensemble are the noisy density curves after adding noise directly through the Fourier coefficient. Aside from §3.5, where parameter estimation is employed, we use the same systems parameters in both the generation of the truth and the data assimilation.

The Eulerian and/or Lagrangian observations are collected and assimilated every one minute, and the entire simulation has 180 total updates. For the Eulerian observation system, we place $N_s = 8$ stationary sensors equidistantly along the ring road. For the Lagrangian observation system, we pick $N_c = 15$ cars with GPS devices that are initially equally spaced along the ring road. The ensemble size is 30 for the EnKF and 300 for the PF because of the high dimension of the state vector.

In each of the following sections, we will outline the traffic scenarios to be studied and the associated macroscopic models, followed by the results of data assimilation in estimating the traffic states which emerge.

3.2 Data assimilation of Eulerian and Lagrangian data

First, we demonstrate that the proposed algorithm can be used to assimilate Eulerian, Lagrangian, and combined Eulerian-Lagrangian observations into a macroscopic model to accurately estimate traffic states. To test the algorithm, we consider a ring road with a traffic light for our simulations and assimilate sensor and GPS data. In order to better evaluate the performance of the algorithm, we will compare the results also to those obtained by simulating the macroscopic model with the same initial data and parameter values but without assimilating observations.

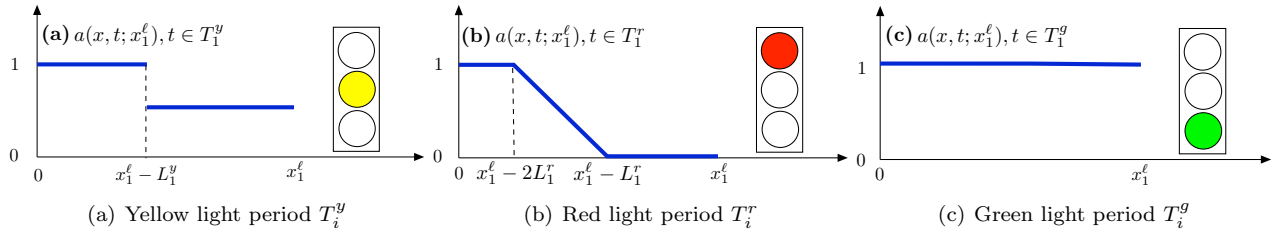


Figure 3: *Deceleration factor* $a(x, t; x_1^\ell)$.

Traffic light scenario: In order to create a dynamic, time-dependent traffic flow, we place a traffic light on the road to force vehicles to decelerate and accelerate periodically, resulting in periodic oscillations of the vehicle density ρ along the road. The resulting model is given by

$$\rho_t + (\rho V_G(\rho(x, t)) a(x, t; x_1^\ell))_x = \epsilon \rho_{xx},$$

$$a(x, t; x_1^\ell) = \begin{cases} 0.5 & (x, t) \in (x_1^\ell - L_1^y, x_1^\ell) \times T_1^y \\ 0 & (x, t) \in (x_1^\ell - L_1^r, x_1^\ell) \times T_1^r \\ (x_1^\ell - L_1^r - x)/L_1^r & (x, t) \in (x_1^\ell - 2L_1^r, x_1^\ell - L_1^r) \times T_1^r \\ 1 & \text{otherwise,} \end{cases} \quad (25)$$

where $x_1^\ell = 25$ miles is the position of the traffic light, and $a(x, t; x_1^\ell)$ is the deceleration rate of the drivers. The function $a(x, t; x_1^\ell)$ is periodic in time t with period T : its graph is shown in Figure 3. The traffic light cycle has a period of $T = 6$ minutes, which consists of yellow light time T_1^y , red light time T_1^r and green light time T_1^g . During yellow lights, drivers within L_1^y miles (see Figure 3(a)) are decelerating. During red lights, cars within L_1^r miles (see Figure 3(b)) are stopped, and those following are decelerating. During green lights T_1^g (see Figure 3(c)), traffic flows normally. We set the lights length $(|T_1^y|, |T_1^r|, |T_1^g|) = (10, 190, 400)$ seconds and response distances $(L_1^y, L_1^r) = (1, 0.8)$ miles in order to generate significantly oscillating traffic flow.

Results: We use the model (25) and the initial conditions and parameters from §3.1 to generate the truth. The same system and parameters are used for the underlying data assimilation model, with perturbed initial conditions.

We carried out simulations using the algorithm described in §2 with the EnKF and PF for (i) Eulerian observations only, (ii) Lagrangian observations only, (iii) both Eulerian and Lagrangian observations, and (iv) no observations. The simulation results based on the traffic lights scenario are shown in Figure 4: panels (a) and (b) contain the results for the EnKF and PF, respectively. The relative RMSE for the cases (i)-(iii) ranges between 1% and 2% after 3 hours, while the relative RMSE for case (iv) (no observations) is above 7%. The results suggest that our algorithm is capable of assimilating any combination of Eulerian and Lagrangian data and performs well independently of whether we use the EnKF or PF.

3.3 Efficacy of Lagrangian data assimilation

In practice, either position or velocity data can be collected through GPS devices. Thus, we are interested in the efficacy and accuracy of different Lagrangian observations. According to the statistical data of GPS devices on the internet, the error of position data has a standard deviation around 5.12 meters [16], while the error of velocity data has a standard deviation around 0.0707 m/s [29, 31]. In this section, both the EnKF and PF are used to assimilate three groups of different Lagrangian observations in a normal traffic scenarios: (1) positions of vehicles, (2) velocities of vehicles, (3) combined positions and velocities of vehicles.

Normal traffic scenario: In this section, we simulate a scenario of unimpeded traffic flow, which we model via the viscous Lighthill-Whitham system (3), with no additional terms added. The initial conditions and parameters are as outlined in §3.1.

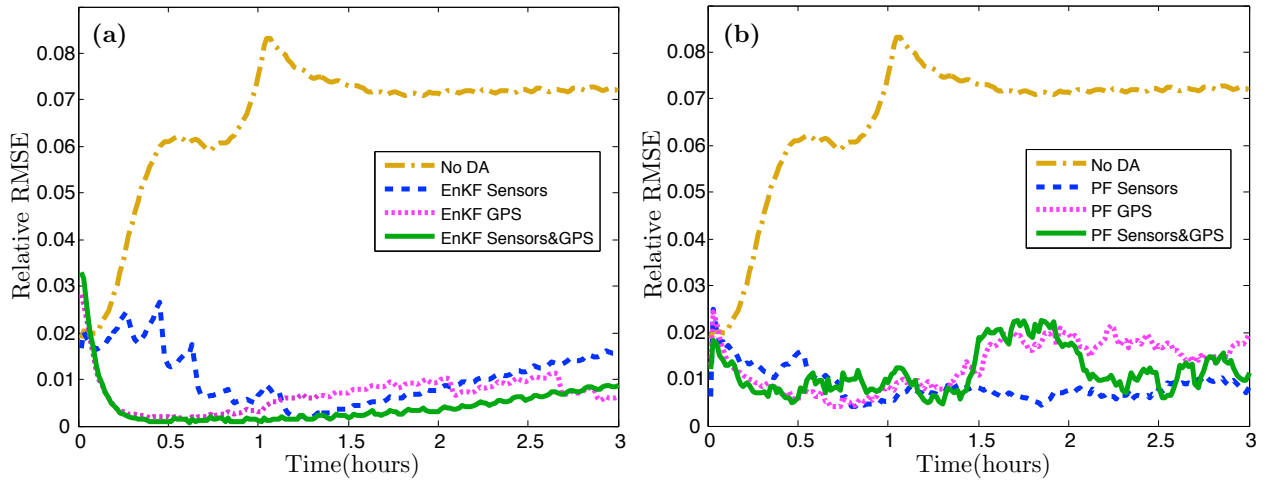


Figure 4: Traffic state estimation in the traffic light scenario (25). Shown is the relative RMSE for assimilating no data (yellow dashed dot), Eulerian data (blue dashed), Lagrangian data (magenta dotted), and both Eulerian and Lagrangian data (green solid) using the (a) EnKF and (b) PF.

Results: The simulation results are shown in Figure 5. Figure 5(a) shows the relative RMSE of Lagrangian data assimilation for different observation data using the EnKF. The error for assimilating position data (solid blue) approaches 3.5%, the error for assimilating velocity data (dashed blue) approaches 0.4% and the error for assimilating both data (dotted blue) approaches 0.1%. In contrast, Figure 5(b) shows the comparison based on the PF: the error for assimilating velocity data (dashed magenta) approaches 0.25%, and the errors for assimilating position data and both data (solid and dotted magenta lines) approach 0.4%. The phenomenon that assimilating both is less accurate initially is related to the “curse of dimensionality” of the PF: the dimension of observations is doubled when assimilating both data, but we are unable to correspondingly increase the ensemble size to compensate.

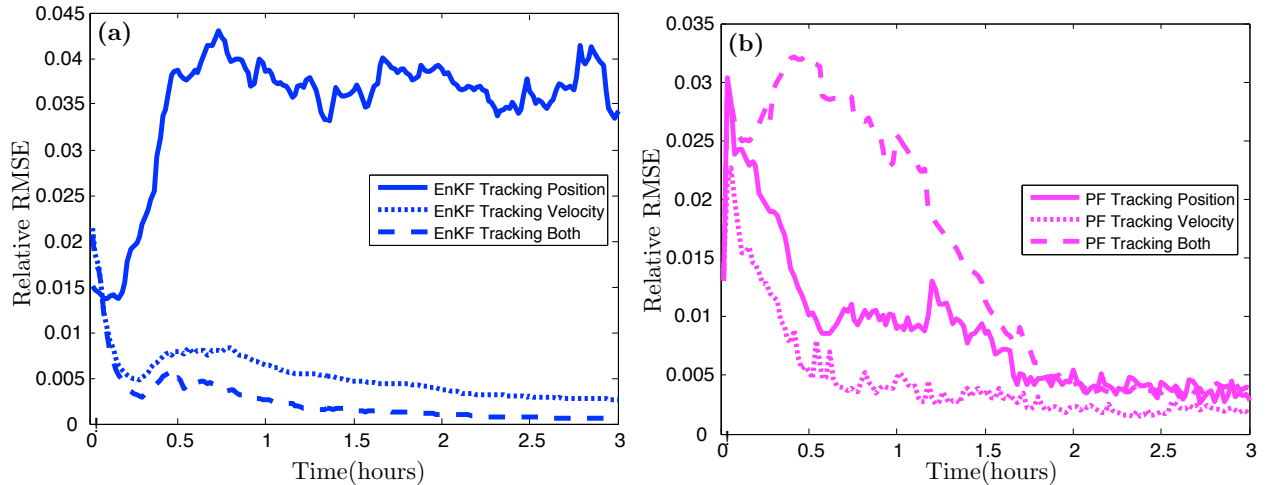


Figure 5: Traffic state estimation comparison for Lagrangian data assimilation based on different observation data from normal traffic flow modeled by (3). Shown is the relative RMSE for the (a) EnKF and (b) PF, where we assimilate position data only (solid), velocity data only (dotted), and combined position and velocity data (dashed).

For both the EnKF and PF, we can see that assimilating velocity data is more accurate compared to assimilating position data, which is consistent with the fact that velocity data has smaller observation noise than position data. However, the PF has a stronger tolerance of observation noise since the difference of error between assimilating velocity data and position data is much smaller compared to those by the EnKF. In addition, the performance

of the EnKF can be improved with combined observation data, while the performance of the PF is limited by the number of particles with combined observation data. Therefore, we will use combined Lagrangian data for EnKF, while only use velocity data for PF in other simulations.

3.4 Impact of sensor location on data assimilation

When the Department of Transportation installs sensors on the road, an important consideration is where to install them in order to optimize the amount of traffic information collected by them. In this section, we are interested in studying the impact of the configuration of sensors in the presence of on-/off-ramps or bottlenecks in order to efficiently place sensors near ramps and bottlenecks. We will design two traffic scenarios for investigation: an on-/off-ramps scenario and a stationary bottleneck scenario.

On/off ramps scenario: We simulate on-ramps and off-ramps by adding source and sink terms to the viscous Lighthill–Whitham equation (3):

$$\rho_t + (\rho V_G(\rho))_x = \epsilon \rho_{xx} + \sum_{i \in \mathcal{I}} \rho_i^{\text{on}}(t) f(x - x_i^{\text{on}}) - \sum_{j \in \mathcal{J}} \rho_j^{\text{off}}(t) f(x - x_j^{\text{off}}), \quad (26)$$

where $\{x_i^{\text{on}}\}_{i \in \mathcal{I}}$ and $\{x_j^{\text{off}}\}_{j \in \mathcal{J}}$ are the positions of on-ramps and off-ramps, and $\{\rho_i^{\text{on}}(t)\}_{i \in \mathcal{I}}$ and $\{\rho_j^{\text{off}}(t)\}_{j \in \mathcal{J}}$ are time-dependent fluxes of on-ramps and off-ramps respectively. In this scenario, we place one on-ramp at $x_1^{\text{on}} = 37.5$ miles and one off-ramp at $x_1^{\text{off}} = 12.5$ miles on the road, and set constant flow $\rho_i^{\text{on}}(t) = 2\rho_{\max}$ for the on-ramp and $\rho_i^{\text{off}}(t) = 4\rho_{\max}$ for the off-ramp. Instead of using a delta function $\delta(x - x_i^{\text{on}})$ for sources and $\delta(x - x_i^{\text{off}})$ for sinks, we introduce sech functions $f(x - x_i^{\text{on}}) = \text{sech}(x - x_i^{\text{on}})$ and $f(x - x_i^{\text{off}}) = \text{sech}(x - x_i^{\text{off}})$ to represent the spread effect of ramps. In this scenario, spatial shocks are generated around the on-ramp and off-ramp. Specifically, a density bump appears around the on-ramp, while a density valley appears around the off-ramp. Three comparative experiments are designed where a sensor is installed (1) at the ramp, (2) 0.5 miles upstream to the ramp, and (3) 0.5 miles downstream to the ramp.

Stationary bottleneck scenario: In this scenario, we aim to model, for instance, construction zones or traffic accidents. By incorporating a “bottleneck factor” into the viscous Lighthill–Whitham equation, we obtain the formula of macroscopic traffic as:

$$\rho_t + (\rho V_G(\rho) a(x - x_1^b))_x = \epsilon \rho_{xx}, \quad (27)$$

where x_1^b is the location of the bottleneck. The quantity $a(x - x_1^b)$ is the bottleneck factor, which can be written as the product of severity coefficient and spread effect $a(x - x_1^b) = cf(x - x_1^b)$. The severity coefficient c reflects the degree of influence of the construction zone or traffic accident, and $f(x - x_1^b)$ represents the spread effect of the bottleneck. We set the bottleneck at $x_1^b = 25$ miles, severity coefficient $c = 1$ and spread effect $f(x) = 1 - 0.5\text{sech}(x)$. In this scenario, a stationary density bottleneck appears around x_1^b . Three comparative experiments are designed where a sensor is installed (1) at the bottleneck, (2) one mile upstream to the bottleneck, and (3) one mile downstream to the bottleneck.

Results: The simulation results for ramps are shown separately in Figure 6(a) and 6(b), and those for the bottleneck are shown in Figure 6(c). In Figure 6(a), it is observed that the performance of data assimilation is improved by moving a sensor from downstream to upstream of the on-ramp. While in Figure 6(b), unlike the on-ramp case, the performance is improved by moving a sensor from upstream to downstream of the off-ramp. In Figure 6(c), it shows that the performance is improved by moving a sensor from upstream to downstream of the bottleneck. By studying the shape of density on the ring road, we find that a density bump is generated gradually upstream to the on-ramp. Therefore, installing a sensor slightly upstream can provide more traffic information and thereby improve the performance of the EnKF. In contrast, a density valley is generated gradually downstream to the off-ramp or the bottleneck, so installing a sensor slightly downstream can provide more traffic information.

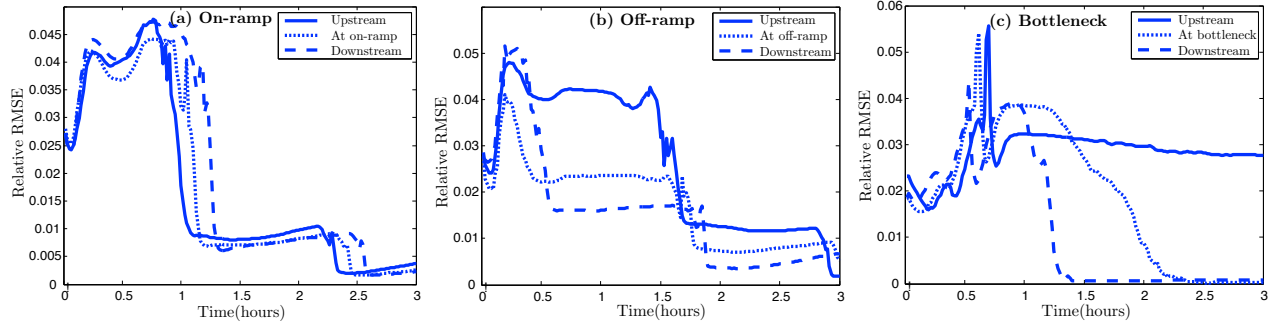


Figure 6: *Evaluation of various sensors configurations using the EnKF. Results for different sensor locations near (a) on-ramps, (b) off-ramps, and (c) bottlenecks are shown, where the solid curves correspond to sensor locations upstream to the target location, the dotted curves represent sensors at the target location, and the dashed curves represent locations downstream to the target location.*

In summary, the locations of sensors do impact the performance of data assimilation when using the EnKF: the simulation results suggest installing sensors upstream of on-ramps and downstream of off-ramps and bottlenecks. In contrast, the PF performs well regardless of the locations of sensors (simulation results not shown).

3.5 Parameter estimation of traffic-flow models

In previous subsections, we showed that our algorithm successfully assimilates traffic states when the parameters in the underlying traffic model are known. However, in practical applications, these parameters are not known and need to be estimated when traffic data are assimilated. Hence, we are interested in the efficacy of data assimilation when traffic parameter estimation is activated. In this section, we consider two traffic scenarios for investigation: an on-/off-ramps scenario and traveling bottleneck scenario. For each scenario, the basic setup for generating the truth and the underlying model for data assimilation is the same as in §3.1; however, the parameters in the underlying traffic model for data assimilation are now taken as unknown.

On/off ramps scenario: The model for the on-/off-ramps scenario is as described in §3.4 for on/off-ramps, and the unknown parameters include maximal density ρ_{\max} , maximal velocity v_{\max} , the fluxes of on-ramp ρ_1^{on} and the fluxes of off-ramp ρ_1^{off} . These parameters are taken to be constant throughout the simulation.

Traveling bottleneck scenario: We also construct a traveling bottleneck scenario, for which we use a model similar to that of the stationary bottleneck in §3.4. To account for a traveling bottleneck, we allow the position of the bottleneck to be time-dependent, and the macroscopic model can be described as:

$$\rho_t + (\rho V_G(\rho) a(x - x_1^b(t)))_x = \epsilon \rho_{xx}, \quad (28)$$

where $x_1^b(t)$ is the location of traveling bottleneck, which is assumed to move back and forth periodically according to the formula $x_1^b(t) = L/2 + L/4 \cos(2\pi t/3)$. The unknown parameters include maximal density ρ_{\max} , maximal velocity v_{\max} , taken to be constant, and the bottleneck location $x_1^b(t)$, which is time-dependent.

Results: As shown in Figure 7(a), the relative RMSE by using the EnKF (blue dashed line) and the PF (magenta dashed line) hovers around 3% and 2%, respectively, after 3 hours of assimilating Eulerian observations. In Figure 7(b), the relative RMSE by using the EnKF (blue dash line) and the PF (magenta dash line) is around 10% and 20%, respectively, after 3 hours of assimilating Lagrangian observations. Observations of the flux, which is the product of density and velocity, can balance the estimation of ρ_{\max} and v_{\max} and thereby provide accurate estimation with an error below 5%. However, Lagrangian observations, especially of velocity data, cannot provide as good a balance between density and velocity as Eulerian observations, and we indeed find that Lagrangian observations are less efficient when the parameters are unknown. We also see in Figure 7(c) and

7(d) that the PF estimator (magenta dashed) is more sensitive when considering the estimated fluxes as these oscillate around the true flux value near the ramps.

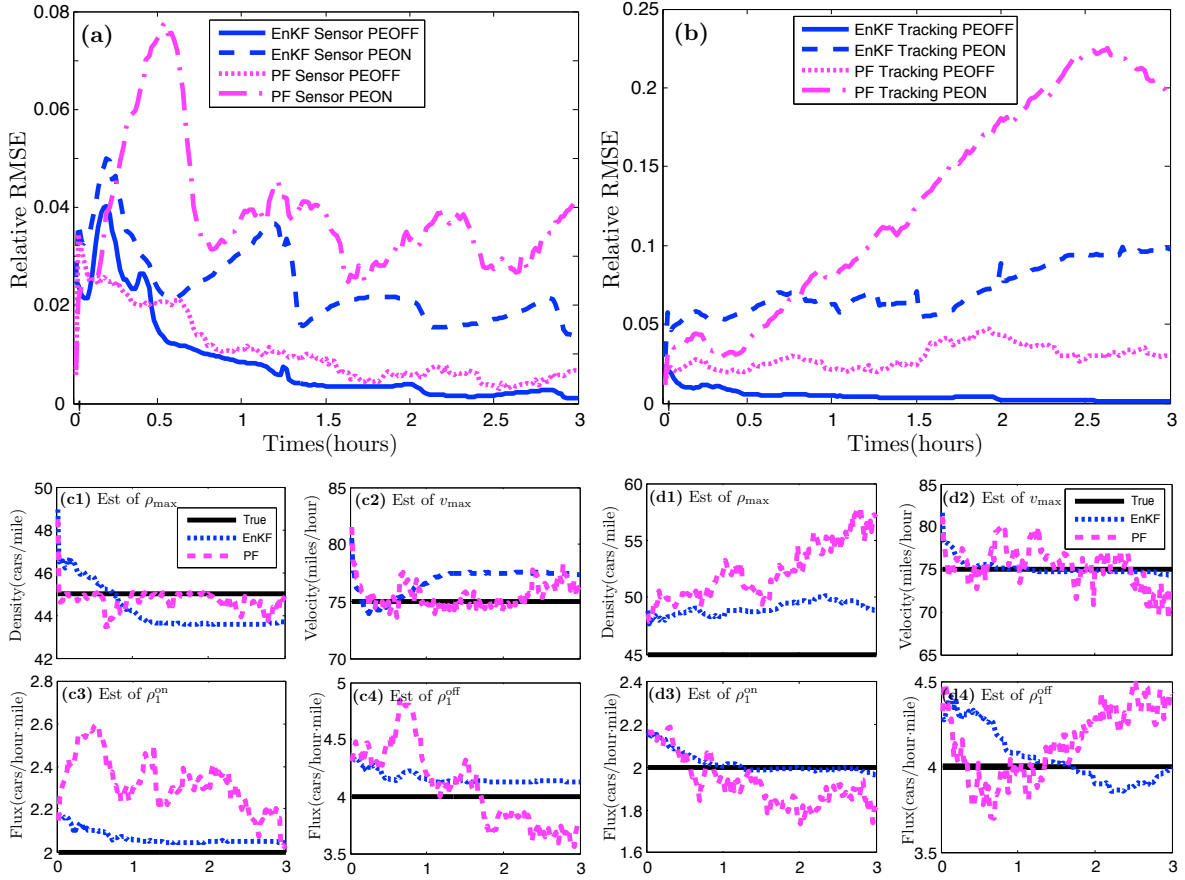


Figure 7: *Estimation of traffic states and parameters in the ramps scenario: shown are the relative RMSE for assimilating (a) Eulerian and (b) Lagrangian observations as well as the results of parameter estimation for (c) Eulerian and (d) Lagrangian observations. Estimated are the maximal density ρ_{max} , the maximal velocity v_{max} , the flux of the on-ramp ρ_1^{on} , and the flux of the off-ramp ρ_1^{off} . PEOFF represents that the parameters are known, while PEON represents that the parameter estimation is considered.*

In the bottleneck scenario, the conclusions are similar as those in the ramps scenario. The relative RMSEs are below 5% in Figure 8(a) and below 9% in Figure 8(b), which means our approach has provided a very good state estimator. In addition, Figure 8(c) and 8(d) demonstrate the dynamical tracking capability of our approach for the time-varying parameter $x_1^b(t)$.

In summary, our approach is capable of providing accurate estimations for both constant and time-varying parameters. Generally Lagrangian observations are more efficient when estimating traffic states, but Eulerian observations appear to be slightly more efficient when parameter estimation is activated.

4 Data assimilation results: truth comes from microscopic models and real data

The simulations in §3 used the same traffic model for (i) generating the truth and (ii) assimilating data in our algorithm. The main goal of this section is to test the implementation of data assimilation to estimate traffic states when the truth is not generated by the underlying model for the data assimilation scheme.

More specifically, in §4.1, we will use data assimilation to estimate traffic states using observations generated

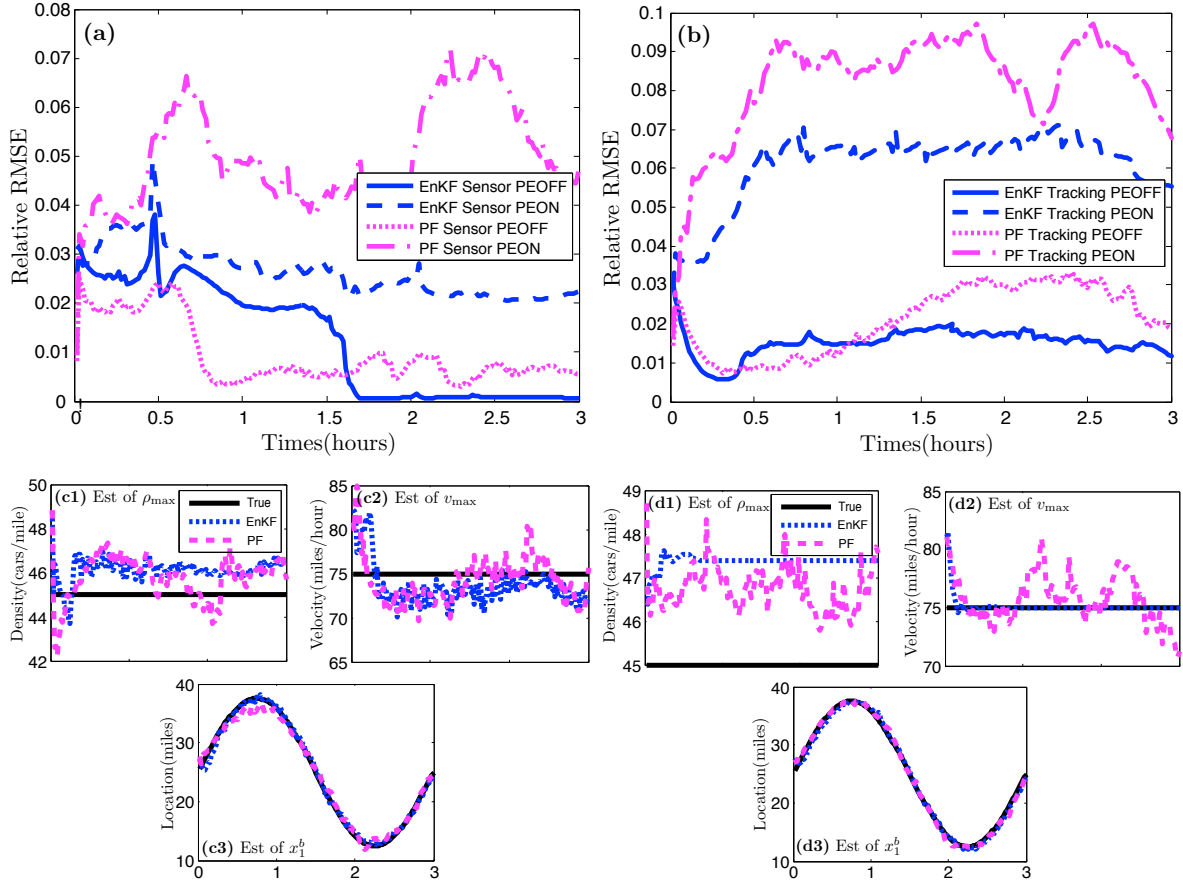


Figure 8: *Estimation of traffic states and parameters in the traveling bottleneck traffic scenario: shown are the relative RMSE for assimilating (a) Eulerian and (b) Lagrangian observations and the results of parameter estimation for (c) Eulerian and (d) Lagrangian observations, where we estimate the maximal density ρ_{max} , the maximal velocity v_{max} , and the time-dependent bottleneck location x_1^b .*

from a microscopic traffic flow model. Then, in §4.2, we will apply parameter estimation to real traffic data obtained from the Minnesota Department of Transportation.

4.1 Truth generated from microscopic models

We are interested in testing the sensitivity with respect to the model used in the data assimilation algorithm in a controlled environment. In this section, we generate synthetic observations using an extended optimal-velocity type microscopic model and assimilate the data using the macroscopic Lighthill–Whitham equation under two different scenarios of highway traffic. These scenarios, to be described in more detail below, are obtained by different choices of the parameters and initial conditions in the microscopic model.

We begin with an overview of the microscopic model, followed by the results when using data assimilation to estimate traffic states in each of the scenarios. In our simulations below, we will study each scenario separately using the EnKF to assimilate Lagrangian observations (vehicle positions) from the microscopic model to estimate the macroscopic traffic density. To compare the predicted traffic densities with the truth, we transform the microscopic traffic data into continuous densities on the ring road.

Microscopic optimal-velocity model: Microscopic models attempt to model traffic through differential equations governing the evolution of the individual cars’ positions and velocities. In general, these models allow for each car to adjust its velocity and acceleration based on the position and/or velocity of its neighbors. We

focus on an extended optimal-velocity type model introduced in [9] in which the drivers adjust their velocity according to not only the headway but also the relative velocity to the car in front. In this model, the position $p_n(t)$ and the individual target headway $s_n(t)$ of the n th vehicle evolve according to

$$\begin{aligned}\tau \ddot{p}_n &= V_g \tanh\left(\frac{p_{n+1} - p_n - s_n}{\ell_0}\right) + V_0 - \dot{p}_n \\ \alpha \dot{s}_n &= \frac{V_g}{\ell_0} (\bar{s} - s_n) - \beta (\dot{p}_{n+1} - \dot{p}_n),\end{aligned}\tag{29}$$

where the dot symbol denotes d/dt . Here τ is the reaction time, and the right hand side of the equation is the so-called optimal velocity function. The adjustment \mathcal{V} depends on the difference of the headway $p_{n+1} - p_n$ from the target headway s_n that encompasses the car length plus a safety distance. A common choice for this function is $\mathcal{V}(u) = \tanh(u)$, which we will use in this paper. Since $\mathcal{V}(0) = 0$, the quantity V_0 represents an optimal velocity at which the car drives when the headway is equal to the optimal headway s_n . The quantity V_g represents a velocity gain, and by considering an infinite headway, we have $\mathcal{V}(\infty) = 1$, so that the quantity $V_g + V_0$ gives an effective speed limit for the drivers. The parameter ℓ_0 is a characteristic length scale which describes the pace at which the driver attempts to achieve the optimal velocity.

The second equation describes the evolution of the individual target headway $s_n(t)$: the term $\bar{s} - s_n$ describes the relaxation of $s_n(t)$ to an optimal headway \bar{s} , while the term $\beta(\dot{p}_{n+1} - \dot{p}_n)$ takes into account that drivers may increase speed if the car in front does so. The parameters α and β are dimensionless: α is a measure of the overall adjustment time of the individual headways, while β measures how proactively drivers react to changes of the relative velocity. Note that setting $(\alpha, \beta) = 0$ recovers the standard optimal-velocity model [7].

We consider the situation of N cars driving on a circular road of fixed length L under the generalized OV model described above which is achieved by adding the periodicity conditions

$$p_{n+N} = p_n + L, \quad s_{n+N} = s_n. \tag{30}$$

It was shown in [9] that the model (29) admits both free-flow and traffic jam (traveling wave) solutions. In this paper, we take the following choice of the parameters $V_g, V_0, \ell_0, \bar{s}, \tau$ which were obtained from Japanese highway data ([6, 13]):

$$V_g = 37.57 \text{ mi/hr}, \quad V_0 = 34.30 \text{ mi/hr}, \quad \ell_0 = 38.15 \text{ ft}, \quad \tau = 0.5 \text{ s}, \quad \bar{s} = 82.00 \text{ ft}, \tag{31}$$

and we consider two different scenarios of highway traffic, to be described below, obtained by different choices of the parameters α, β, N, L and initial conditions. In the first scenario, a square wave evolves into a shock with gradually decreasing amplitude (a solution that can also qualitatively be found in the macroscopic Lighthill–Whitham model), while the second scenario describes the emergence of a traveling wave with fixed amplitude (which cannot be generated by the Lighthill–Whitham model).

Results (square wave scenario): The first scenario is that of a long circular road consisting of a large number of cars engaged in free flow interacting with a small traffic jam region. We set $(\alpha, \beta) = (1, 4)$ and the length $L = 22$ miles. To initialize the positions of the cars, we place cars equally spaced along the road with a fixed density of 40 cars per mile except for a two mile stretch in which the density is increased to 130 cars per mile for a total of $N = 1060$ cars. This gives that the initial headways are $s_0^f = 131.95$ ft in the free flow region and $s_0^j = 40.61$ ft in the traffic jam region. The initial velocities of the cars are taken to be

$$v_0^f = V_g \tanh\left(\frac{s_0^f - \bar{s}}{\ell_0}\right) + V_0, \quad v_0^j = V_g \tanh\left(\frac{s_0^j - \bar{s}}{\ell_0}\right) + V_0, \tag{32}$$

for the free flow and jam regions, respectively. We initialize the system with the above conditions and compute the local density at each time step using a circular kernel density estimator in Matlab. In terms of density, the initial condition resembles a square wave which evolves into a shock with decreasing amplitude over time (Figure 9).

The simulation results for the first scenario are shown in Figure 9. We observe that the estimated density (green circled line) gradually converges to the true value (blue starred line). This is plausible as the viscous Lighthill–Whitham model can generate a traveling shock with decreasing amplitude as demonstrated in §2.1. With the aid of data assimilation, the macroscopic model provides an accurate estimation for the traveling shock wave transferred from microscopic traffic data.

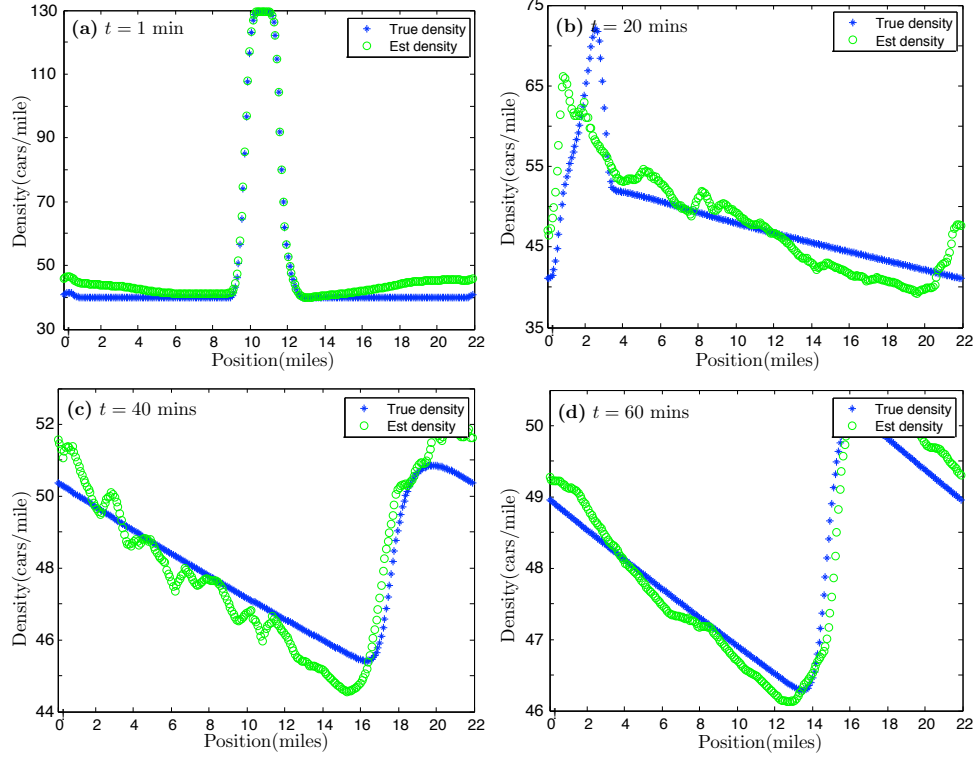


Figure 9: *Estimating microscopic data in scenario 1 using the EnKF. True and estimated density values are represented by blue pluses and green circles, respectively.*

Results (traveling wave scenario): The second scenario describes the propagation of a traffic jam or traveling wave solution. We set $(\alpha, \beta) = (4, 0.01)$ and the length $L = 2.5$ miles. We place $N = 150$ cars offset from an equal spacing of length $(n - 1)L/N$ on the road with initial positions and velocities

$$p_n(0) = \frac{(n - 1)L}{N} + 10 \sin \left(\frac{2\pi(n - 1)}{N} \right), \quad v_n(0) = V_g \tanh \left(\frac{s_0 - \bar{s}}{\ell_0} \right) + V_0, \quad (33)$$

where $s_0 = 87.97$ ft is the initial average headway. Figure 10 shows the evolution of the local density of this system. After a short time, a traffic jam, or traveling wave, emerges that travels backwards against the flow of traffic.

Also shown in Figure 10 are the results of the simulation using data assimilation. We observe that the estimated density (green circles) is smoothed out rapidly and does not converge to the true density value (blue pluses). Unlike the previous case, the macroscopic model cannot provide an accurate estimation even though data assimilation is used. This is not unexpected as the viscous Lighthill–Whitham model cannot generate a traveling square wave shape with stable amplitude and is therefore not capable of reproducing the correct density profile.

In summary, a macroscopic model may produce inaccurate results if the model is not capable of generating dynamical features present in the traffic flow data that need to be assimilated.

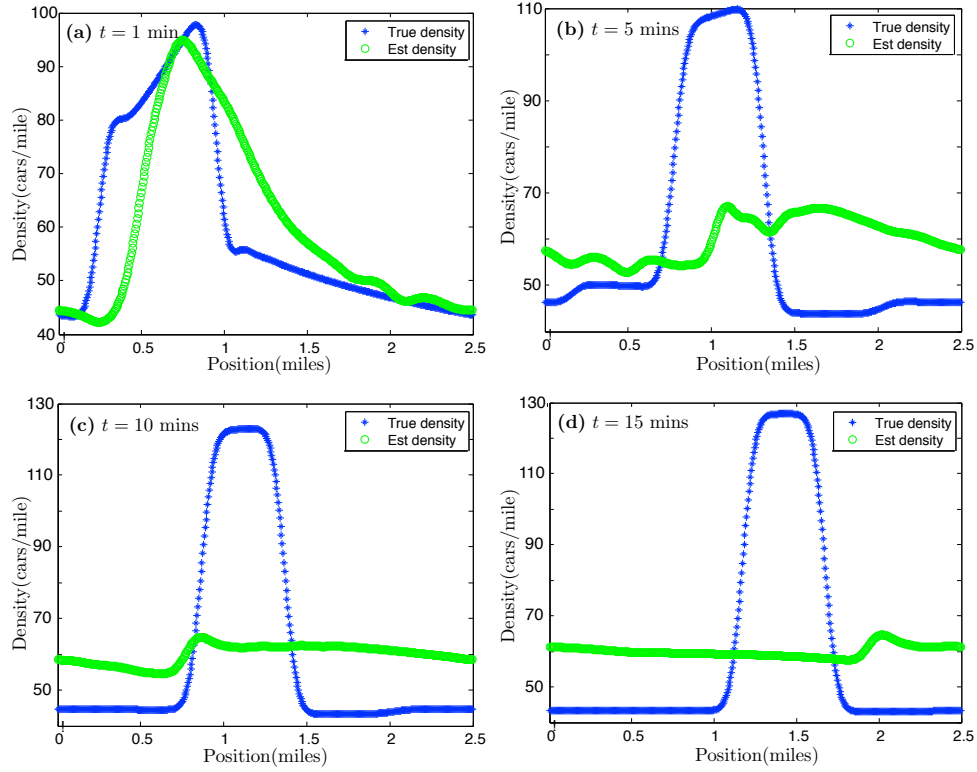


Figure 10: *Estimating microscopic data in scenario 2 using the EnKF. True and estimated density values are represented by blue pluses and green circles, respectively.*

4.2 Parameter estimation for real traffic data

In addition to estimating simulated traffic states and parameters, we are interested in applying the developed approach to real traffic data. In this section, we use data from the Minnesota Department of Transportation [30].

Traffic scenario: A freeway strip of I-35E with length 7.175 miles between Main Street and Hwy 96E is considered. There are 14 consecutive detectors S1535-S1548 along this stretch, as well as three on-ramps and one off-ramp (see Figure 11). We picked two representative time periods from a day for simulation: one is late night (00:01am-00:30am) when free flow is expected, and the other is rush hour (05:31pm-06:00pm) when traffic jams are expected. The Minnesota Department of Transportation provides traffic density, velocity and flow data collected from the sensors, and the density data of ramps. All traffic data are collected each minute.

We use a macroscopic model that takes the on- and off-ramps into account, and assimilate the flux data collected from the sensors using the EnKF. We also estimate the unknown maximal density ρ_{\max} and the maximal velocity v_{\max} . The density profiles of the ramps are artificially taken as unknown and are replaced by the average density value during a period. In contrast to the simulation presented in previous sections, we use an adaptive inflation algorithm [2] here.

Results: The simulation results for late night traffic are shown in Figure 12. The relative RMSE by assimilating all observation approaches 15%, while the relative RMSE by not assimilating observations approaches 40%. The parameter estimation in 12(b) shows that estimated maximal velocity \hat{v}_{\max} is approximately 85 miles/hour, and the maximal density $\hat{\rho}_{\max}$ decreases to below 30 cars/mile. This is a reasonable description of late night traffic.

The simulation results for rush hour traffic are shown in Figure 13. The approach results in a decrease of error from 15% to 8%. Since the flow of ramps is quite stable during rush hour, the average density value for ramps

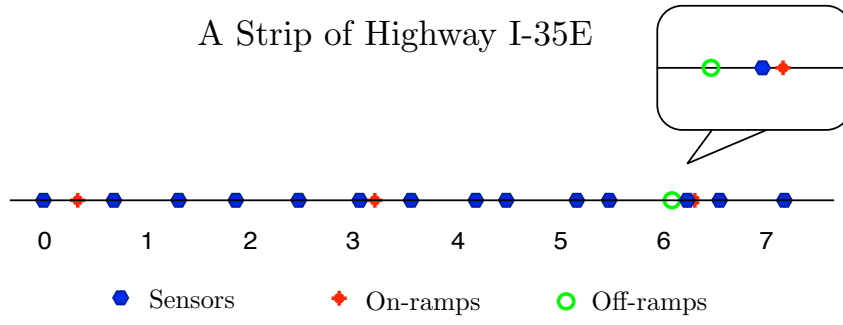


Figure 11: A strip of highway I-35E in Minnesota. Cars are moving from left to right. Sensors, on-ramps, and off-ramps are labelled using blue stars, red pluses, and green circles, respectively.

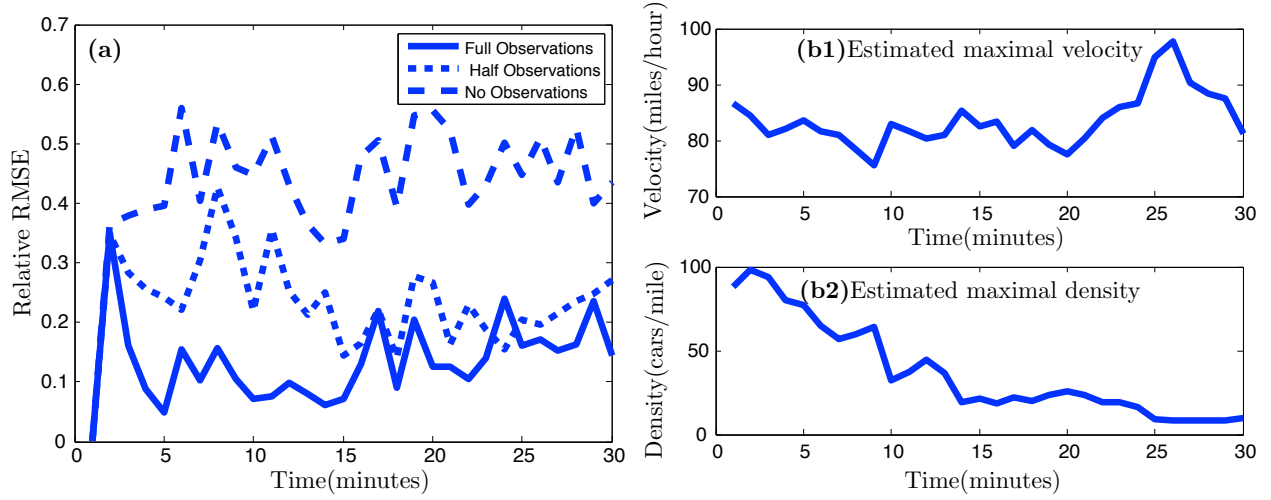


Figure 12: Relative RSME (a) and parameter estimate (b) are shown for data assimilation of real traffic data taken during late night. Observations are taken from all sensors (solid), half the sensors (dotted), or no sensors (dashed).

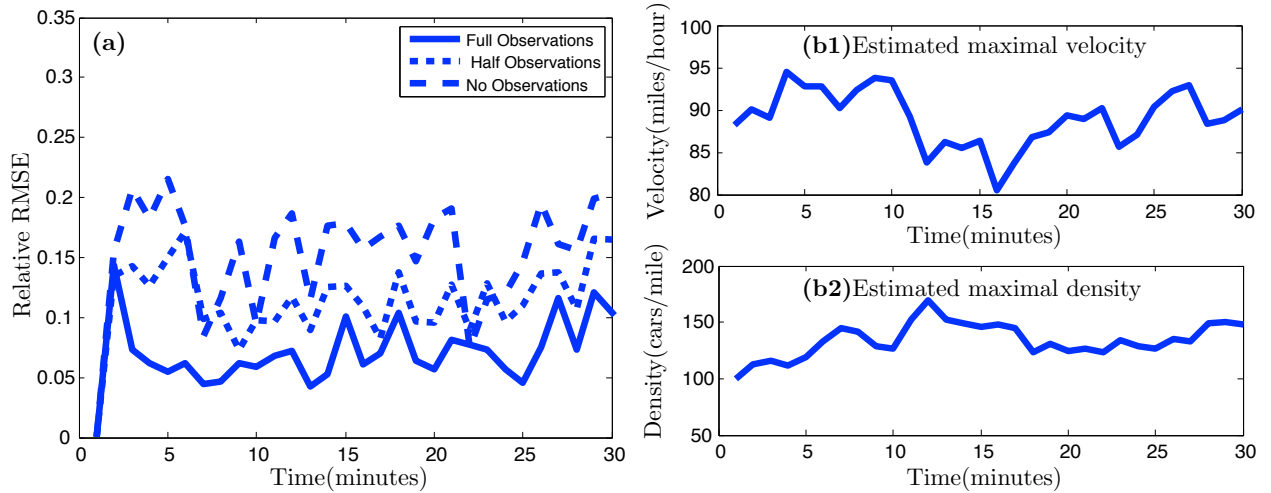


Figure 13: Relative RSME (a) and parameter estimate (b) are shown for data assimilation of real traffic data collected during rush hour. Observations are taken from all sensors (solid), half the sensors (dotted), or no sensors (dashed).

are close to the true density value which explains why our approach results in a less drastic improvement when compared to the previous case. In Figure 13(b), the estimated maximal velocity \hat{v}_{\max} is still approximately 85 miles/hour, but the maximal density $\hat{\rho}_{\max}$ increases to 150 cars/mile, which is characteristic of traffic flow during rush hour.

In summary, the efficacy of our approach has been validated by the real traffic data from Minnesota Department of Transportation. Estimation for traffic states is significantly improved in both late night and rush hour conditions, and the estimation for parameter is consistent with characteristics of traffic scenarios.

5 Conclusions and Discussion

In this paper, we studied several aspects of traffic flow prediction: mathematical traffic models, observation models, and data assimilation techniques. Specifically, we reviewed the Lighthill-Whitham macroscopic model that is used as the basic underlying traffic model in data assimilation (§2.1). We then discussed two types of observations: Eulerian sensor data and Lagrangian GPS data, and developed a formulation that is capable of assimilating the Eulerian and Lagrangian observations simultaneously (§2.2). We also investigated how our approach could be used to estimate traffic states as well as parameters using the EnKF and PF (§2.3, §2.4).

The initial motivation of this paper was to propose an efficient approach that could assimilate both Eulerian sensor data and Lagrangian GPS data simultaneously. The idea behind our approach is to append the differential equations for the positions and velocities of the vehicles to the macroscopic traffic model in order to solve them simultaneously. Unlike the previous methods, our approach is capable of handling flux data, position data and velocity data efficiently without reformulation in Lagrangian coordinates or reshaping the macroscopic traffic model. This approach has been shown to accurately estimate traffic states in different traffic scenarios with true underlying traffic model (§3). We also studied how the choice of filters and observations affect data assimilation: compared to the EnKF, the PF is less sensitive to observation noise and sensor locations, but its computation cost is higher (§3.3, §3.4); the positions of sensor locations impact the accuracy of the EnKF, and relevant suggestions for installing sensors are provided (§3.4). However, this approach is less accurate when the true traffic model is unknown and an estimated underlying traffic model is used (§4).

One limitation we reported on in §4 is that the accuracy of data assimilation is reduced significantly if the underlying traffic model cannot reproduce the actual traffic flow from which observations are collected: in such a situation, the model error is very large and will therefore dominate the overall error. Hence, to increase accuracy in such cases requires the development of better, more accurate traffic flow models that are capable of reproducing a wider range of traffic flows.

Another limitation is that the proposed approach for assimilating Lagrangian observations requires knowledge of the velocity of a vehicle at a given position: in our macroscopic model for the density, the velocity was given explicitly as a function of the density, which allowed us to access the velocity in our algorithm. If the underlying traffic-flow model does not provide information about the velocity of vehicles at any specified position, our method for assimilating Lagrangian observations will not work.

There are various extensions that could be implemented to improve the framework presented here. First, the proposed data assimilation approach was only applied to simplified traffic scenarios such as on and off ramps, traffic lights, and bottlenecks that were implemented in the Lighthill-Whitham equation. In practice, the road network is typically more complicated and may include, for instance, intersections and multiple lanes. An exciting extension would be to expand the above framework to more complex and realistic traffic scenarios. Second, it would be interesting to see whether the algorithm could be extended to use parameter estimation to relay traffic information to drivers. For instance, the period of traffic lights could be estimated by data assimilation, and thus the waiting time could be provided to drivers. Data assimilation could also be used to predict the location and duration of congestion caused by slow trucks or road constructions.

Acknowledgements The authors gratefully acknowledge support by the National Science Foundation through grant DMS-1148284. We thank the anonymous referees for their constructive and helpful comments.

References

- [1] Federal Highway Administration, *Over-roadway sensor technologies*, <http://www.fhwa.dot.gov/policyinformation/pubs/vdstits2007/05pt2.cfm>.
- [2] J. L. Anderson, *An adaptive covariance inflation error correction algorithm for ensemble filters*, *Tellus* **59A** (2007), 210–224.
- [3] J. L. Anderson, *Spatially and temporally varying adaptive covariance inflation for ensemble filters*, *Tellus A* **61** (2009), no. 1, 72–83.
- [4] J. L. Anderson and S. L. Anderson, *A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts*, *Mon. Wea. Rev.* **127** (1999), no. 12, 2741–2758.
- [5] S. J. Baek, B. R. Hunt, E. Kalnay, and E. Ott, *Local ensemble Kalman filtering in the presence of model bias*, *Tellus* **58A** (2006), no. 3, 293–306.
- [6] M. Bando, K. Hasebe, K. Nakanishi, and A. Nakayama, *Analysis of optimal velocity model with explicit delay*, *Phys. Rev. E* **58** (1998), no. 5, 5429.
- [7] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama, *Dynamical model of traffic congestion and numerical simulation*, *Phys. Rev. E* **51** (1995), 1035–1042.
- [8] G. Burgers, P. J. van Leeuwen, and G. Evensen, *Analysis scheme in the ensemble Kalman filter*, *Mon. Wea. Rev.* **126** (1998), 1719–1724.
- [9] P. Carter, P. Christiansen, Y. Gaididei, C. Gorria, B. Sandstede, M. Sorensen, and J. Starke, *Multi-jam solutions in traffic models with velocity-dependent driver strategies*, *SIAM J. Appl. Math.* **74** (2014), no. 6, 1895–1918.
- [10] A. Doucet, N. De Freitas, and N. Gordon, *An introduction to sequential Monte Carlo methods*, *Sequential Monte Carlo methods in practice*, Springer, 2001, pp. 3–14.
- [11] A. Doucet, S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, *Stat. Comput.* **10** (2000), 197–208.
- [12] G. Evensen, *The ensemble Kalman filter: theoretical formulation and practical implementation*, *Ocean dynamics* **53** (2003), no. 4, 343–367.
- [13] Y. B. Gaididei, C. Gorria, R. Berkemer, A. Kawamoto, T. Shiga, P. L. Christiansen, M. P. Sørensen, and J. Starke, *Controlling traffic jams by time-modulating the safety distance*, *Phys. Rev. E* (2013).
- [14] G. Gaspari and S. E. Cohn, *Construction of correlation functions in two and three dimensions*, *Q. J. Roy. Meteor. Soc.* **125** (1999), no. 554, 723–757.
- [15] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, *Novel approach to nonlinear-non-Gaussian Bayesian state estimation*, *IEE Proc-F* **140** (1993), no. 2, 107–113.
- [16] GPS Gov, *GPS accuracy*, <http://www.gps.gov/systems/gps/performance/accuracy/>.
- [17] B. D. Greenshields, W. Channing, H. Miller, et al., *A study of traffic capacity*, Highway research board proceedings, vol. 1935, National Research Council (USA), Highway Research Board, 1935.
- [18] T. M. Hamill, J. S. Whitaker, and C. Snyder, *Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter*, *Mon. Wea. Rev.* **129** (2001), no. 11, 2776–2790.
- [19] W. K. Hastings, *Monte Carlo sampling methods using Markov Chains and their applications*, *Biometrika* **57** (1970), 97–109.
- [20] D. Helbing, *Traffic and related self-driven many-particle systems*, *Rev. of Mod. Phys.* **73** (2001), no. 4, 1067.

- [21] P. Henrici, *Applied and computational complex analysis, discrete Fourier analysis, Cauchy integrals, construction of conformal maps, univalent functions*, vol. 3, John Wiley & Sons, 1993.
- [22] P. L. Houtekamer and H. L. Mitchell, *Data assimilation using an ensemble Kalman filter technique*, Mon. Wea. Rev. **126** (1998), 796–811.
- [23] P. L. Houtekamer and H. L. Mitchell, *A sequential ensemble Kalman filter for atmospheric data assimilation*, Mon. Wea. Rev. **129** (2001), no. 1, 123–137.
- [24] A. Kong, J. Liu, and W. Wong, *Sequential imputations and Bayesian missing data problems*, J. Am. Stat. Assoc. **89** (1994), 278–288.
- [25] M. J. Lighthill and G. B. Whitham, *On kinematic waves. ii. a theory of traffic flow on long crowded roads*, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **229** (1955), no. 1178, 317–345.
- [26] J. Liu and M. West, *Combined parameter and state estimation in simulation-based filtering*, Sequential Monte Carlo methods in practice, Springer, 2001, pp. 197–223.
- [27] J. Mandel., *Efficient implementation of the ensemble kalman filter*, University of Colorado at Denver and Health Sciences Center, Center for Computational Mathematics, 2006.
- [28] R. N. Miller, M. Ghil, and F. Gauthiez, *Advanced data assimilation in strongly nonlinear dynamical systems*, J. Atmos. Sci. **51** (1994), no. 8, 1037–1056.
- [29] GPS Information Net, *GPS speed*, <http://www.gpsinformation.net/main/gpsspeed.htm>.
- [30] Minnesota Department of Transportation, *Traffic forecasting & analysis*, <http://www.dot.state.mn.us/traffic/data/>.
- [31] GPS Action Replay, *GPS and speed measurement*, <http://gpsactionreplay.free.fr/index.php?menu=6&choice=5>.
- [32] P. I. Richards, *Shock waves on the highway*, Oper. Res. **4** (1956), no. 1, 42–51.
- [33] H. Salman, L. Kuznetsov, C. K. R. T. Jones, and K. Ide, *A method for assimilating Lagrangian data into a shallow-water-equation ocean model*, Mon. Wea. Rev. **134** (2006), no. 4, 1081–1101.
- [34] C. Snyder, Proceedings of the ECMWF Seminar on Data Assimilation for Atmosphere and Ocean, 2011.
- [35] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, *Obstacles to high-dimensional particle filtering*, Mon. Wea. Rev. **136** (2008), no. 12, 4629–4640.
- [36] C. Snyder, T. Bengtsson, and M. Morzfeld, *Performance bounds for particle filters using the optimal proposal*, Mon. Wea. Rev. **143** (2015), no. 11, 4750.
- [37] Lloyd N. Trefethen, *Spectral methods in MATLAB*, SIAM, 2000.
- [38] P. J. van Leeuwen, *Particle filtering in geophysical systems*, Mon. Wea. Rev. **137** (2009), 4089–4114.
- [39] D. B. Work, S. Blandin, O.-P. Tossavainen, B. Piccoli, and A. M. Bayen, *A traffic model for velocity data assimilation*, Appl. Math. Res. Express **2010** (2010), no. 1, 1–35.
- [40] D. B. Work, O.-P. Tossavainen, S. Blandin, A. M. Bayen, T. Iwuchukwu, and K. Tracton, *An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices*, 2008 47th IEEE Decis. Contr. P., IEEE, 2008, pp. 5062–5068.
- [41] Y. Yuan, H. Van Lint, F. Van Wageningen-Kessels, and S. Hoogendoorn, *Network-wide traffic state estimation using loop detector and floating car data*, J. Intell. Transport. S.: Technology, Planning, and Operations **18** (2014), no. 1.
- [42] Y. Yuan, J. W. C. Van Lint, R. Eddie Wilson, F. Van Wageningen-Kessels, and S. P. Hoogendoorn, *Real-time Lagrangian traffic state estimator for freeways*, IEEE Trans. Intell. Transp. Syst **13** (2012), no. 1, 59–70.